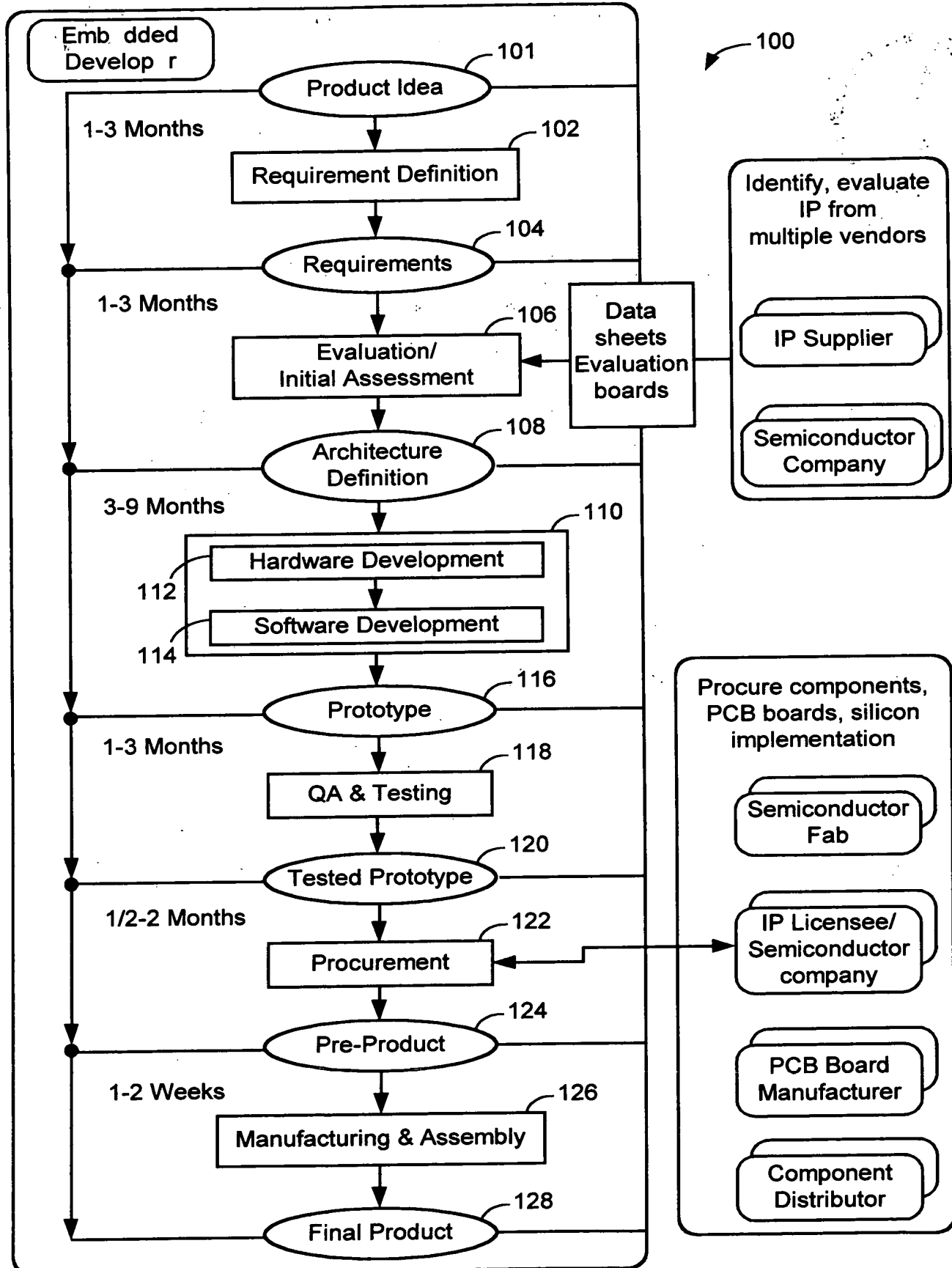


#4



Figur 1 (Prior Art)

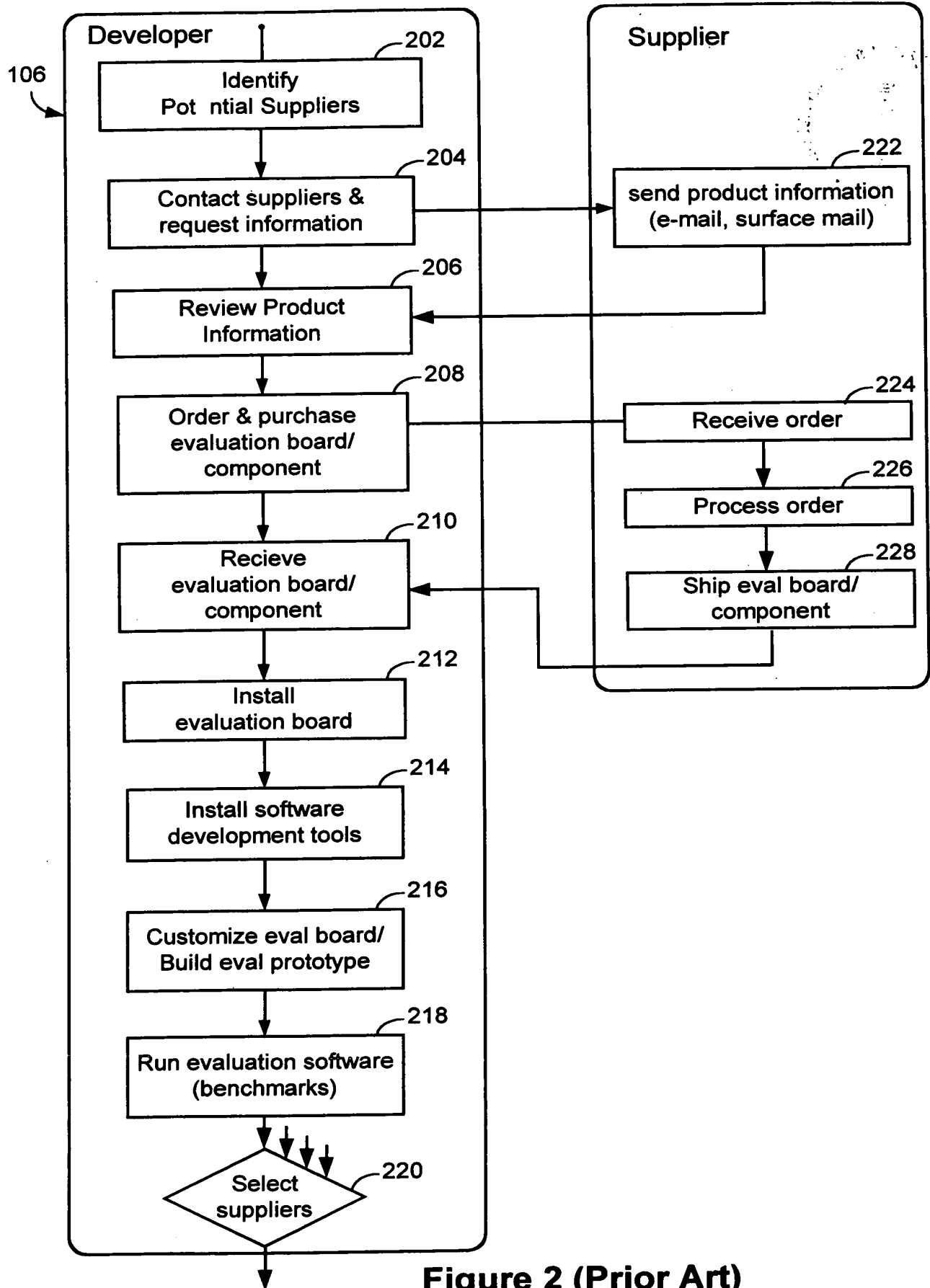
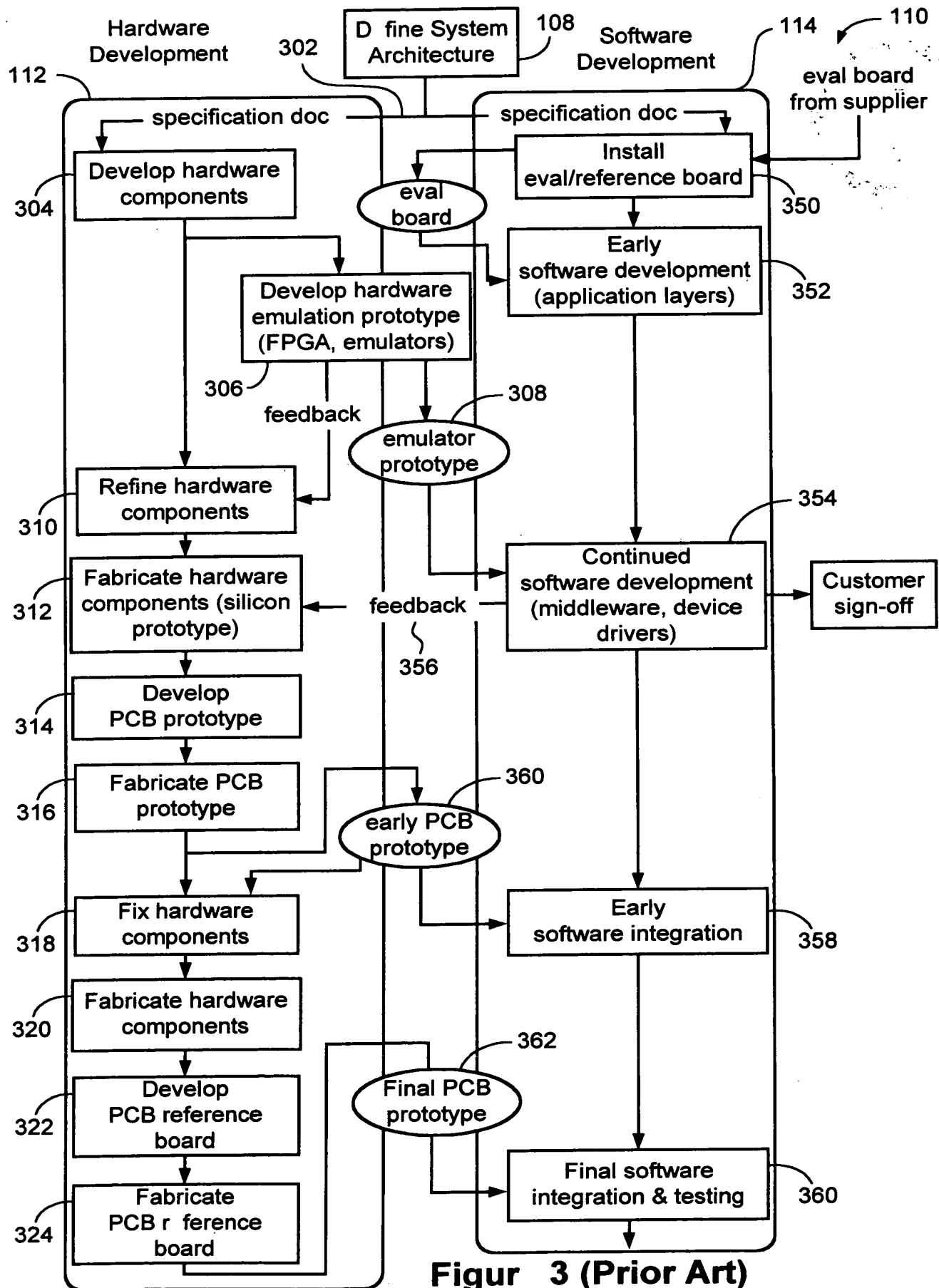


Figure 2 (Prior Art)



Figur 3 (Prior Art)

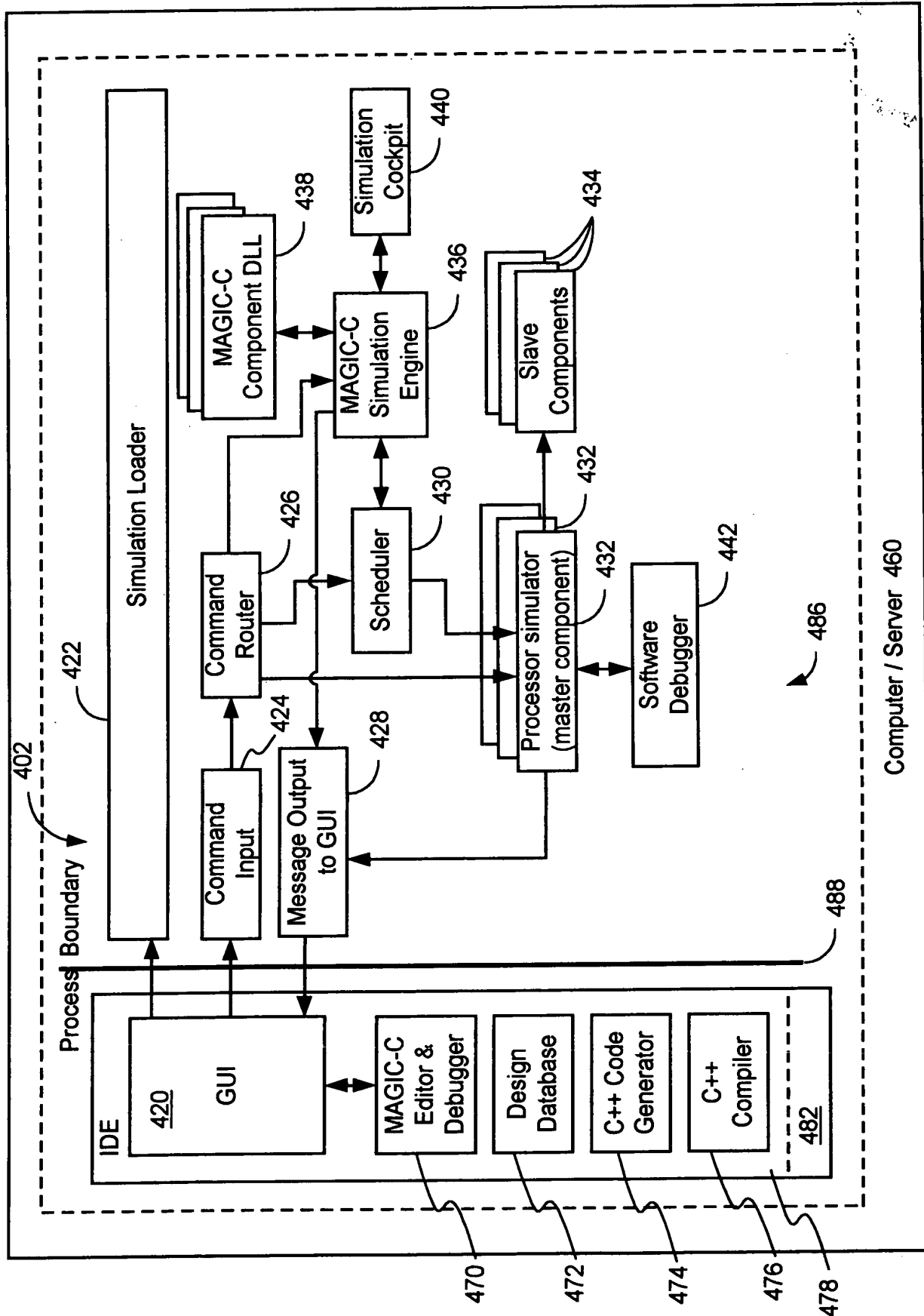


Figure 4A

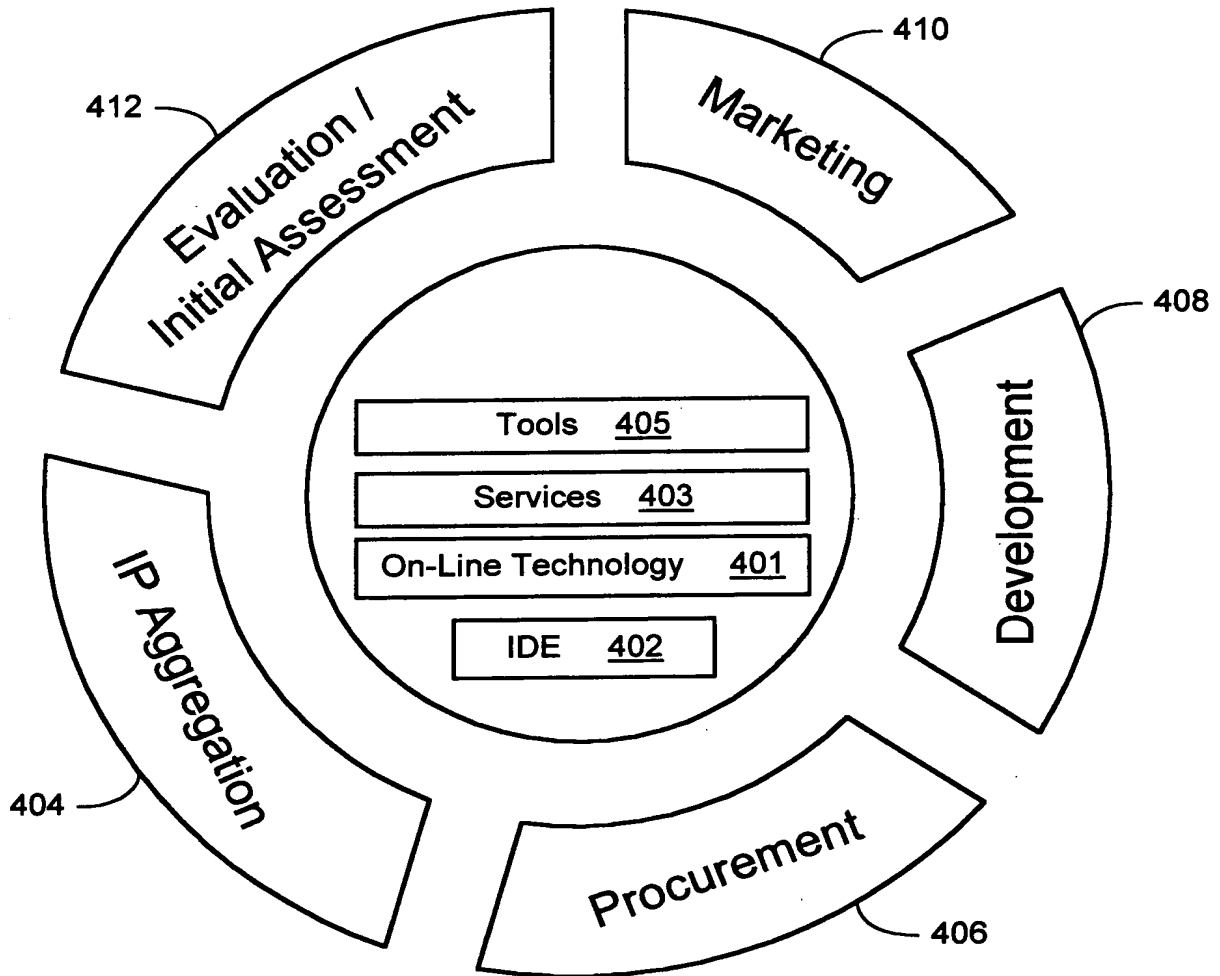
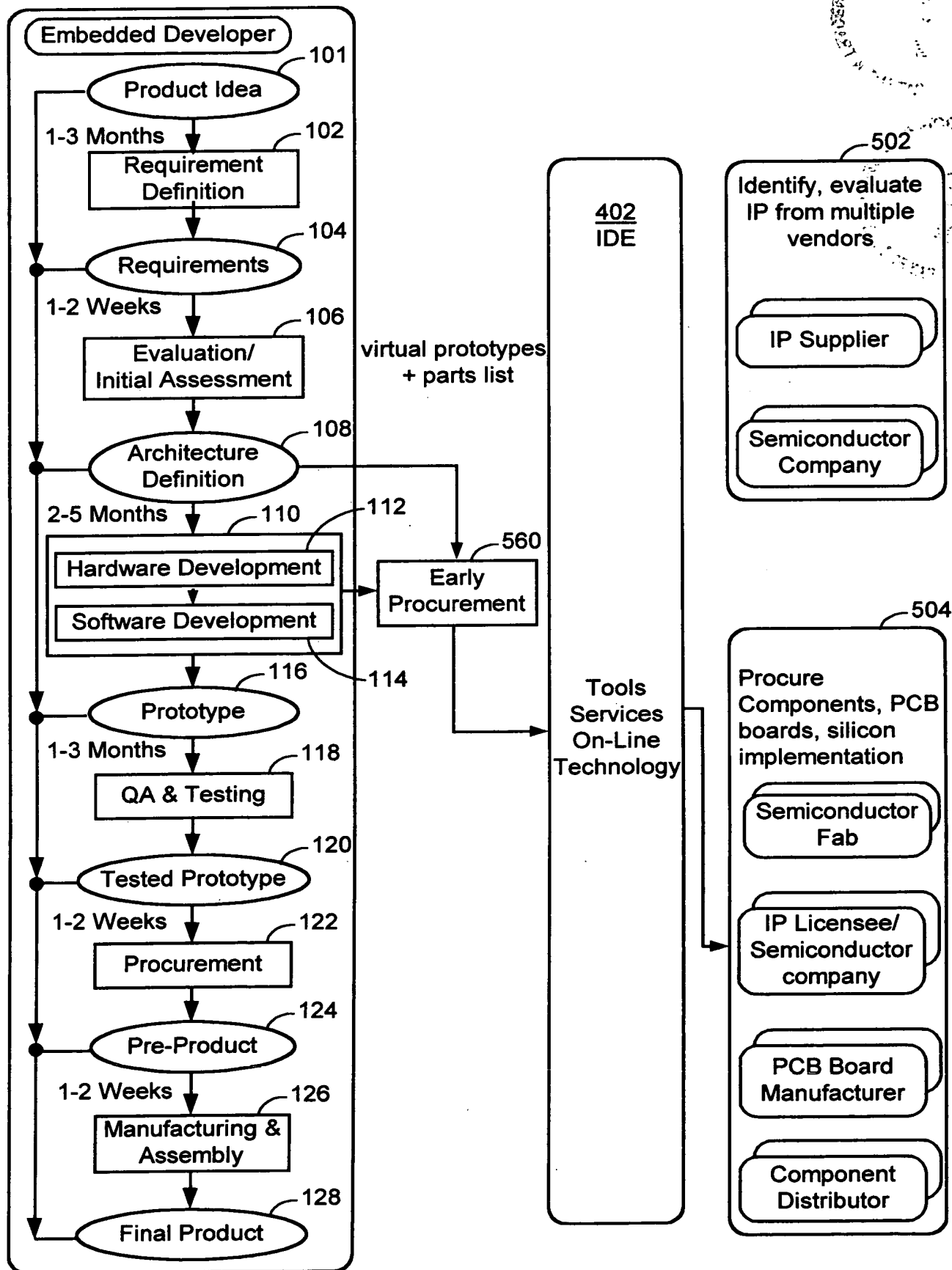
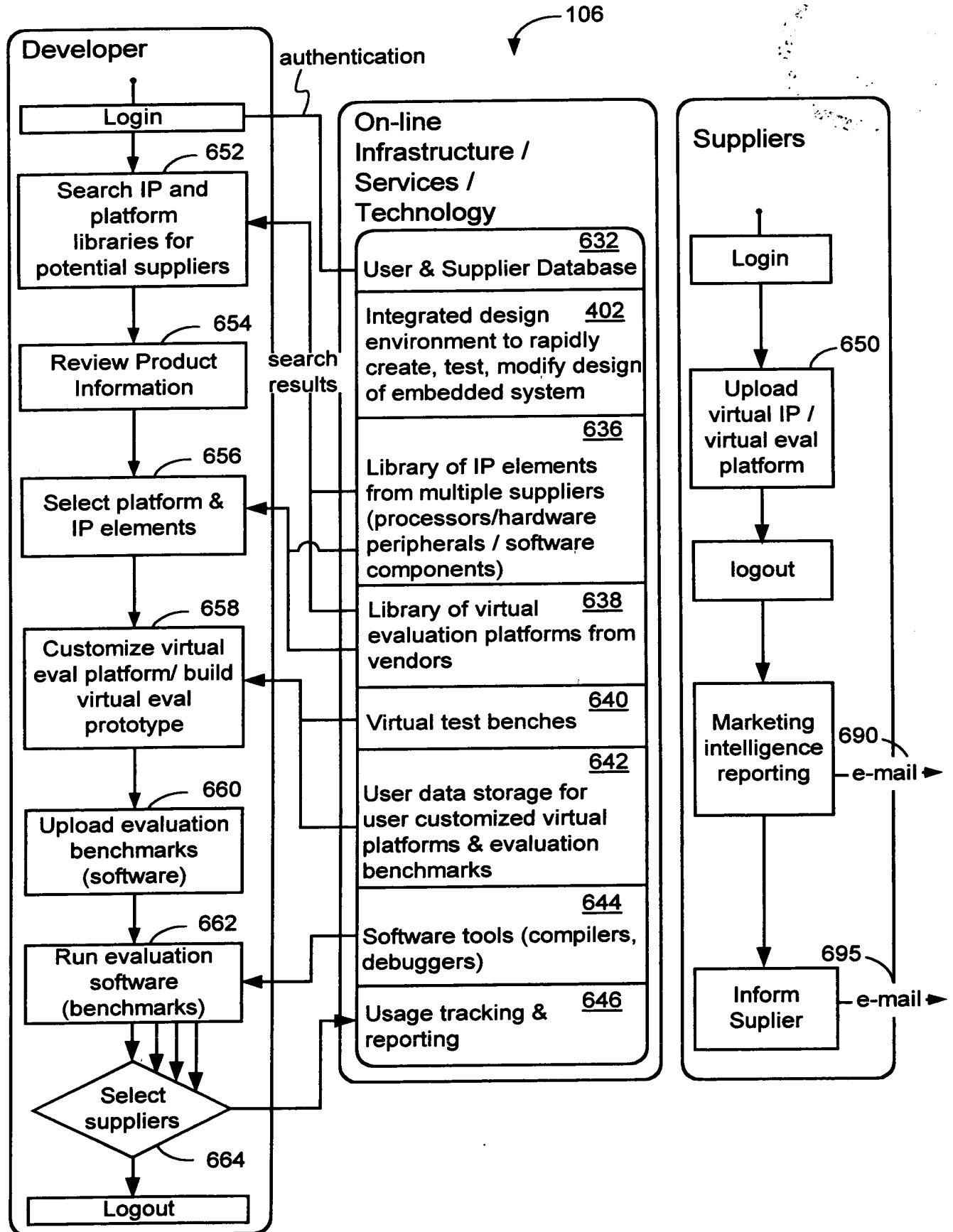


Figure 4B



Figur 5



Figur 6

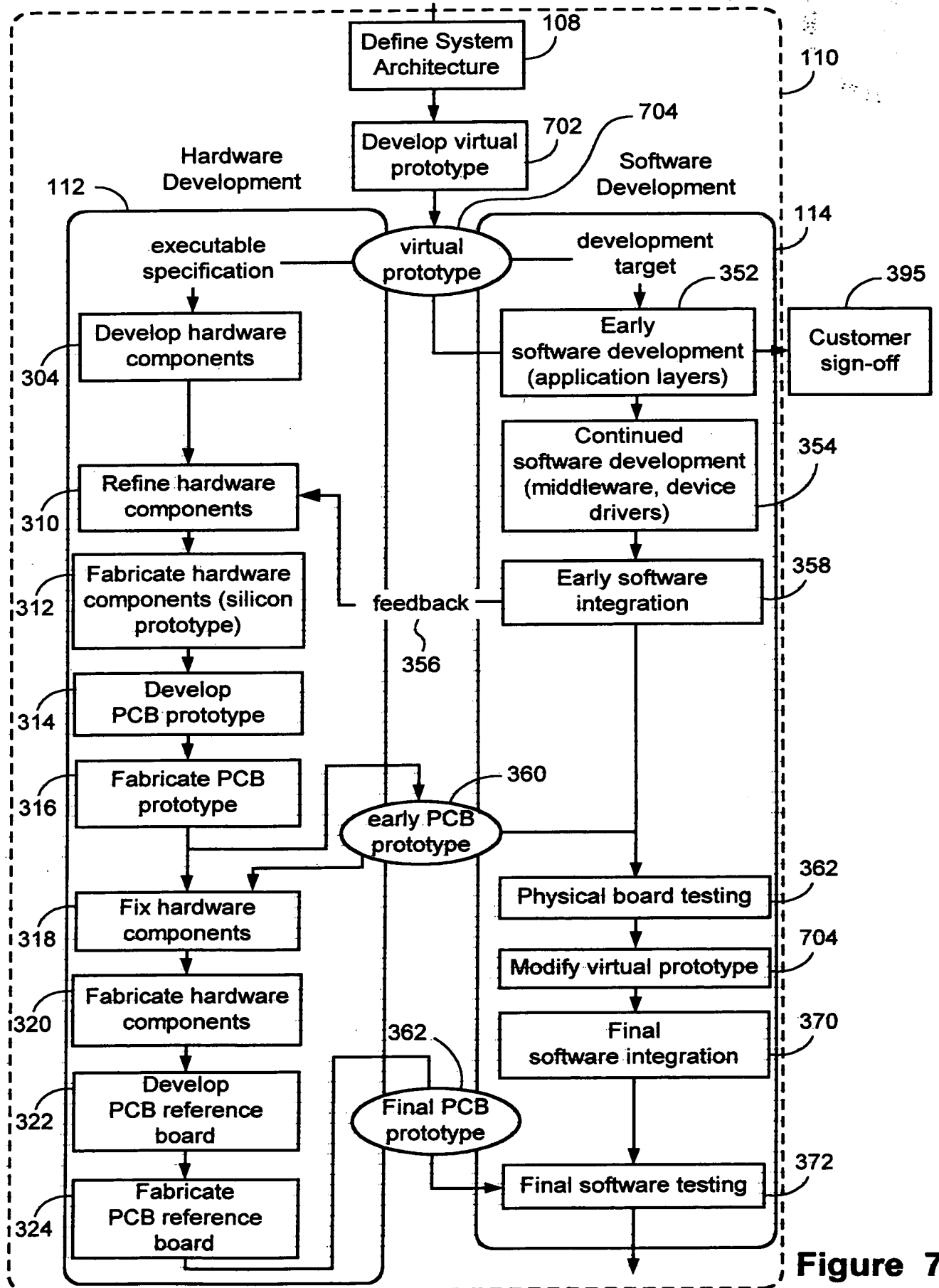


Figure 7

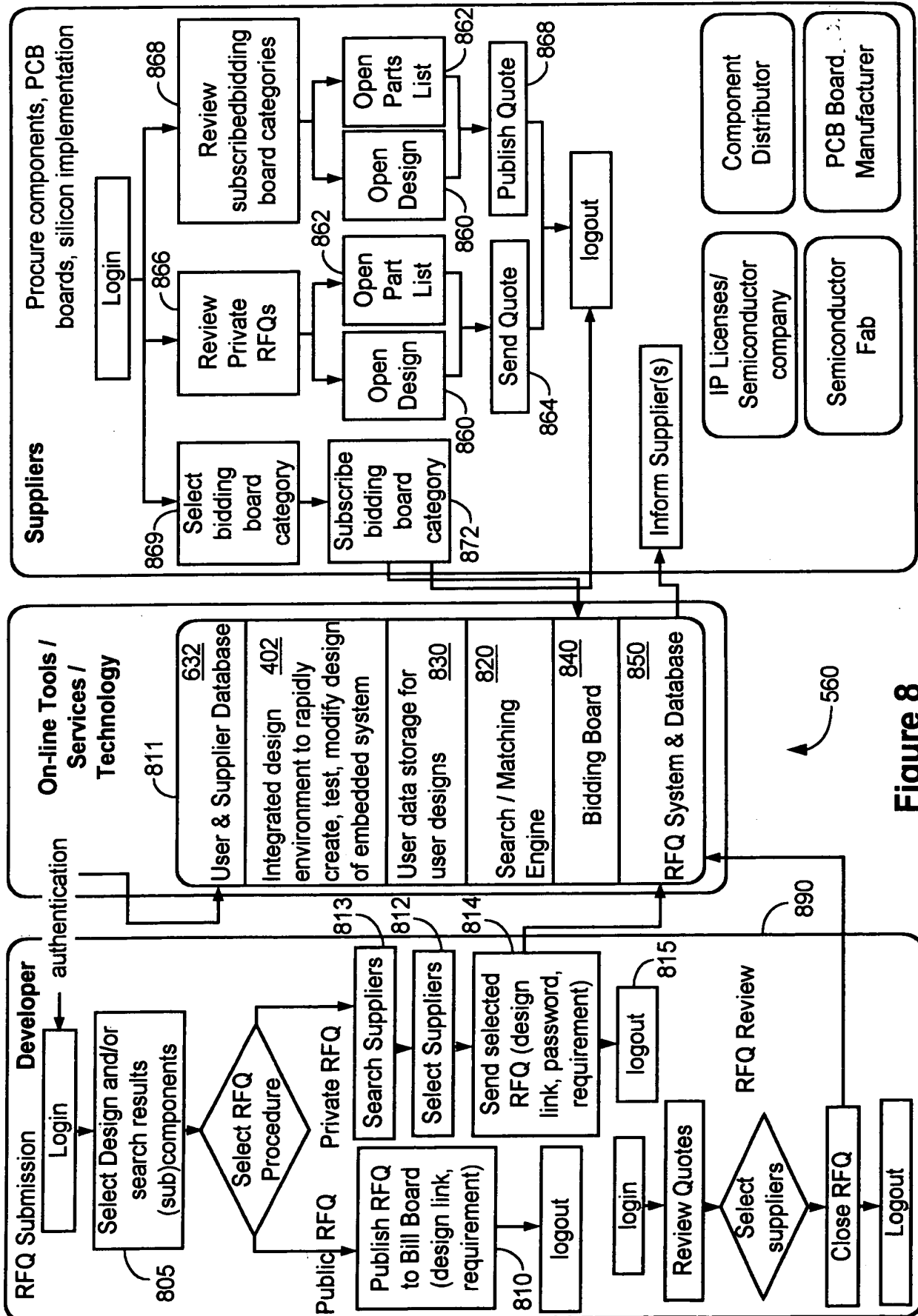


Figure 8

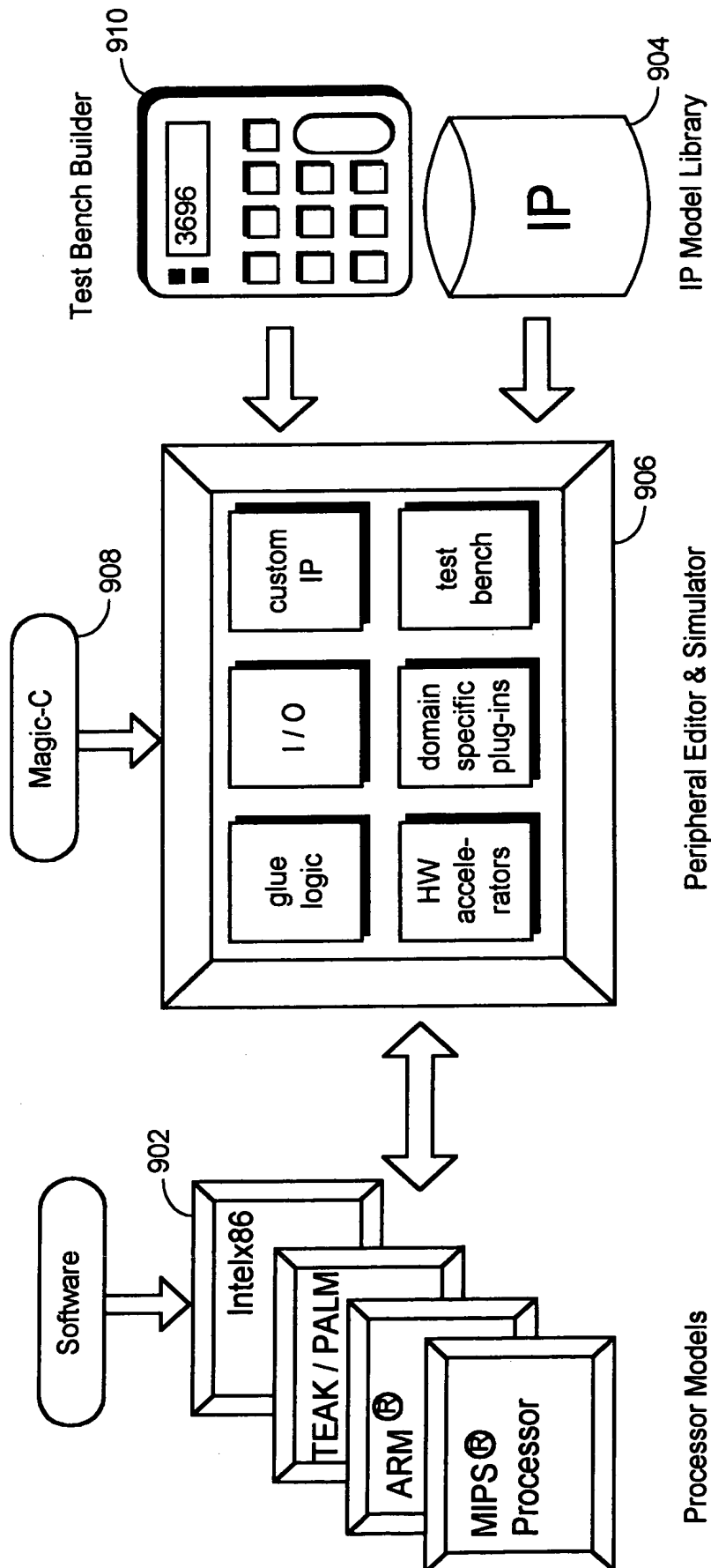
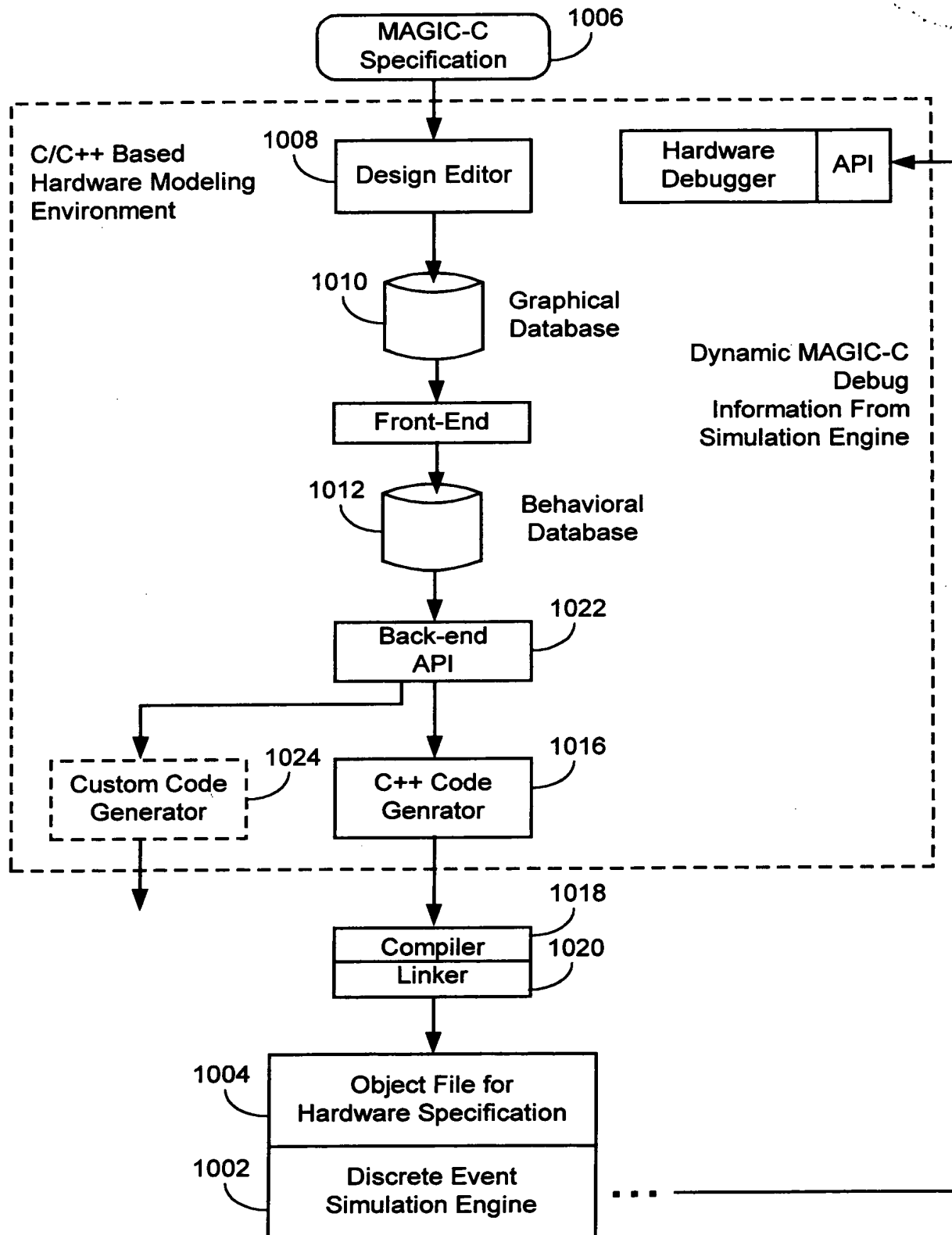


Figure 9



Figur 10

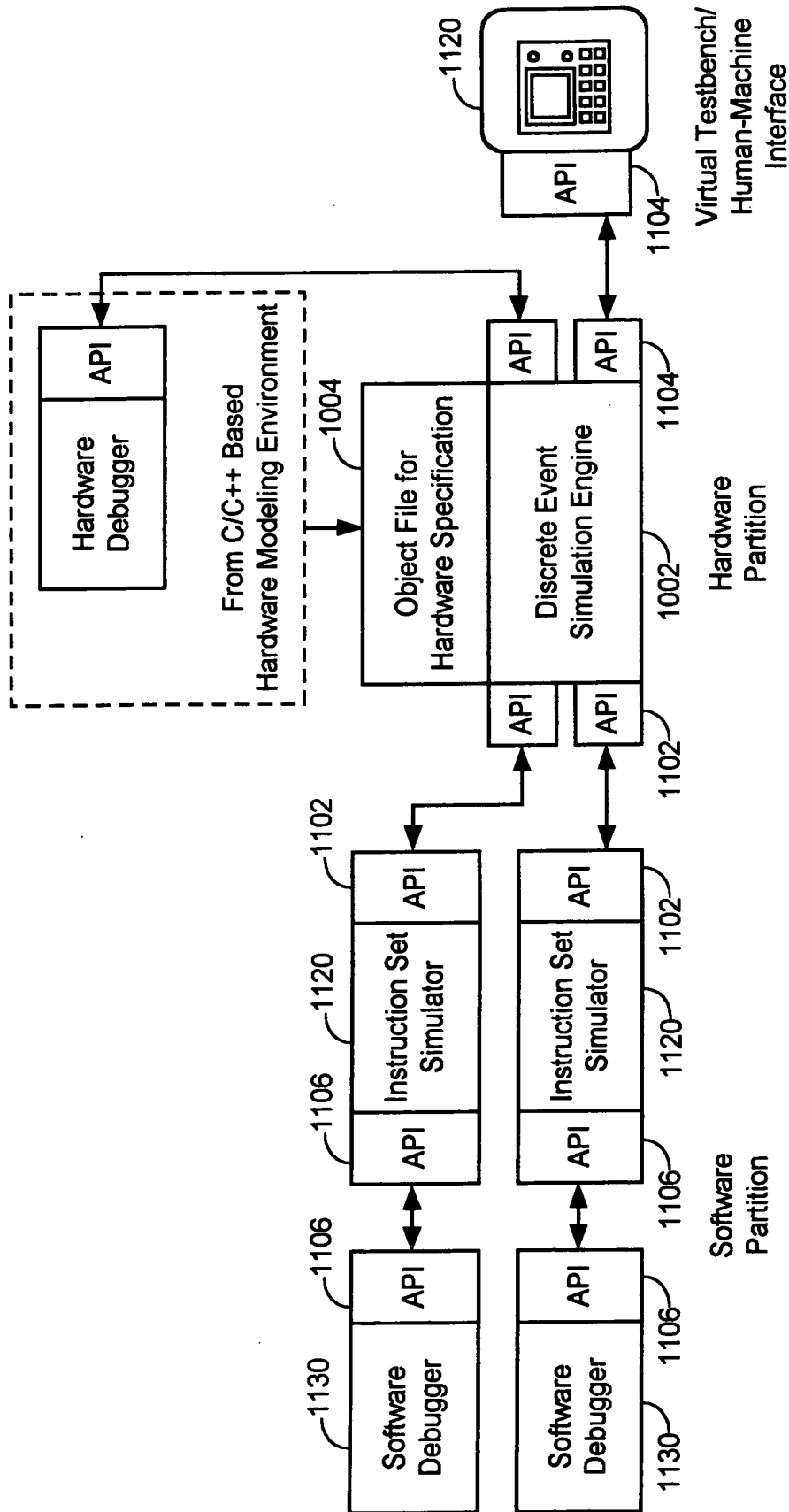
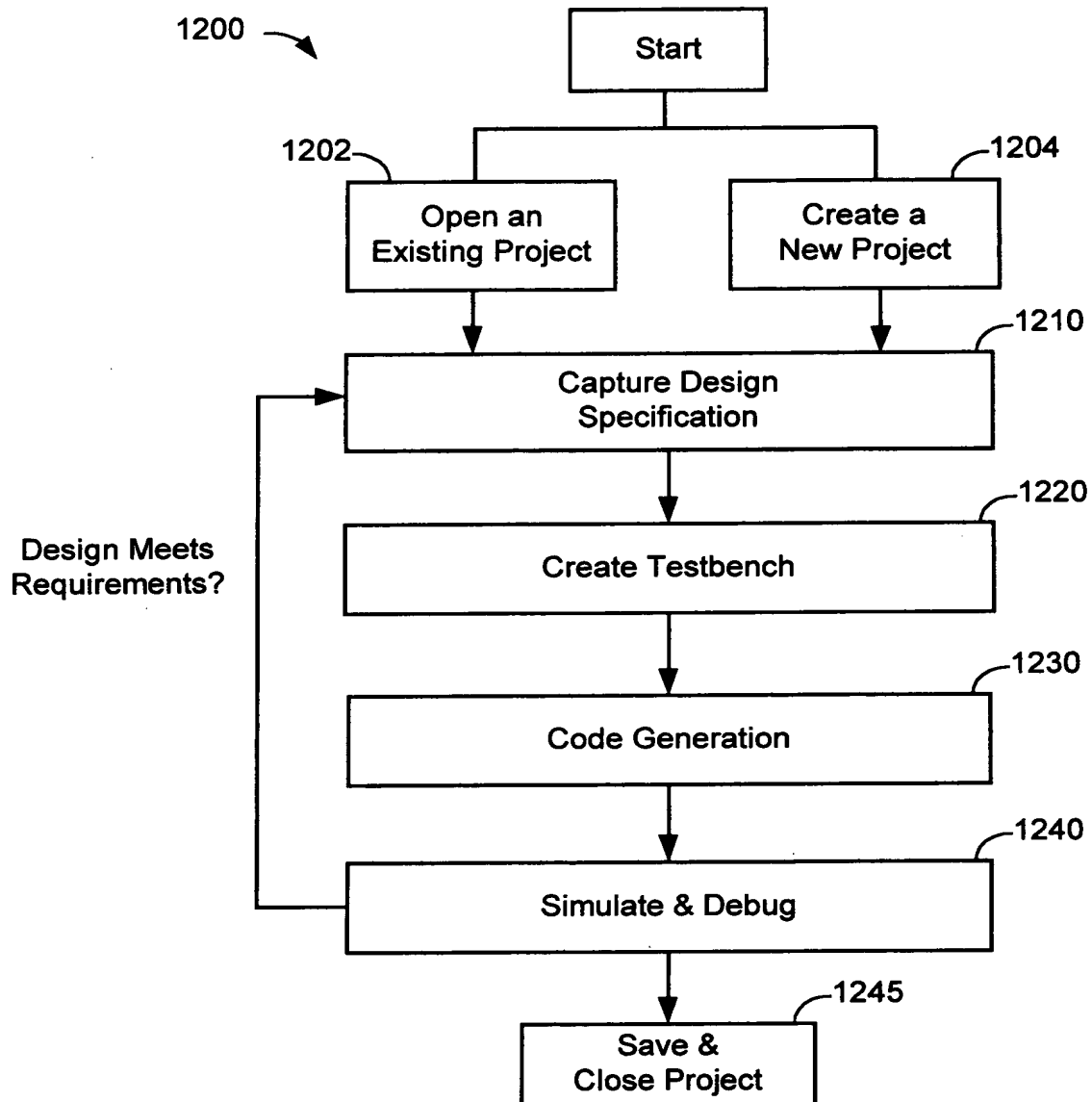
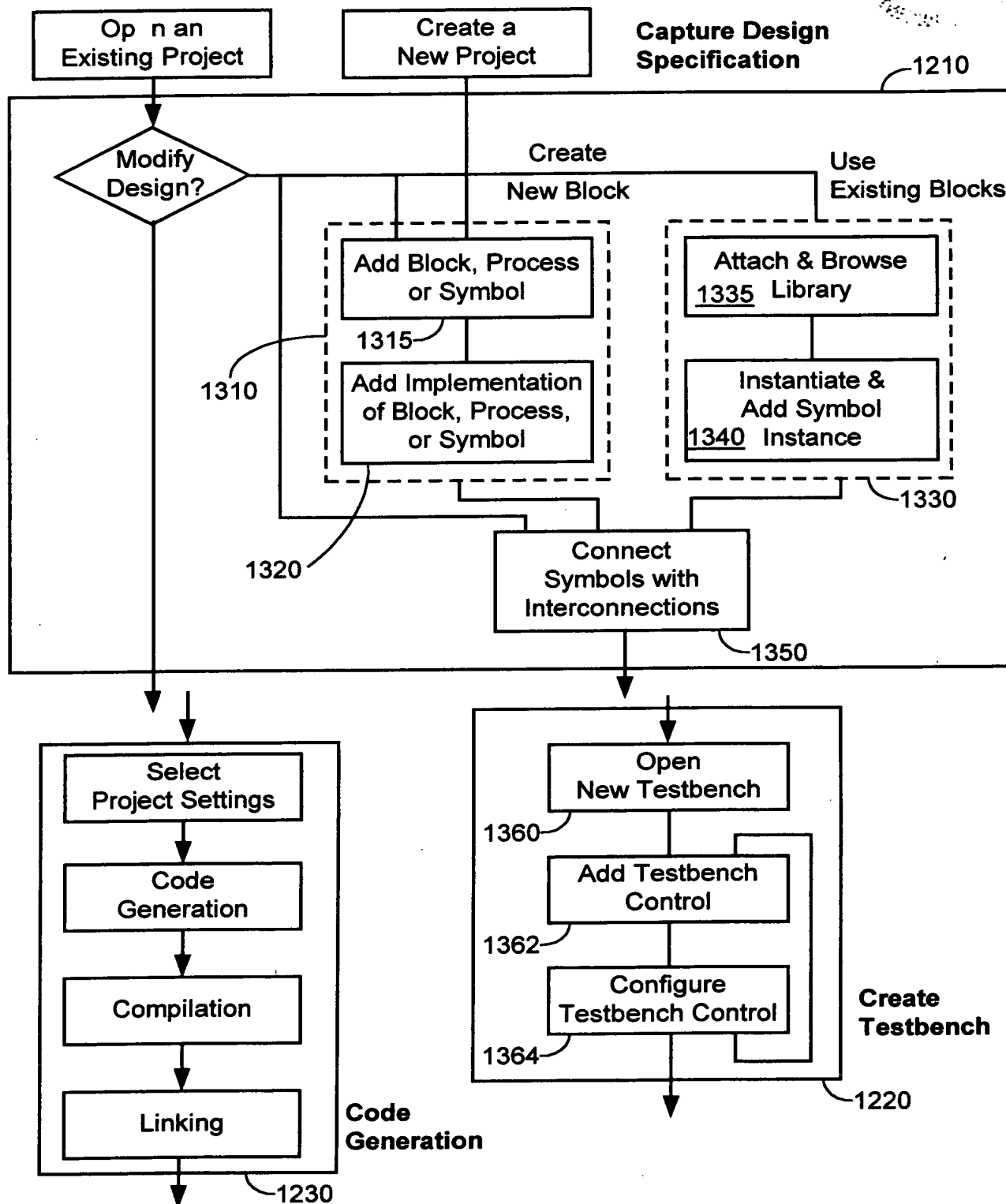


Figure 11



Figur 12



Figur 13

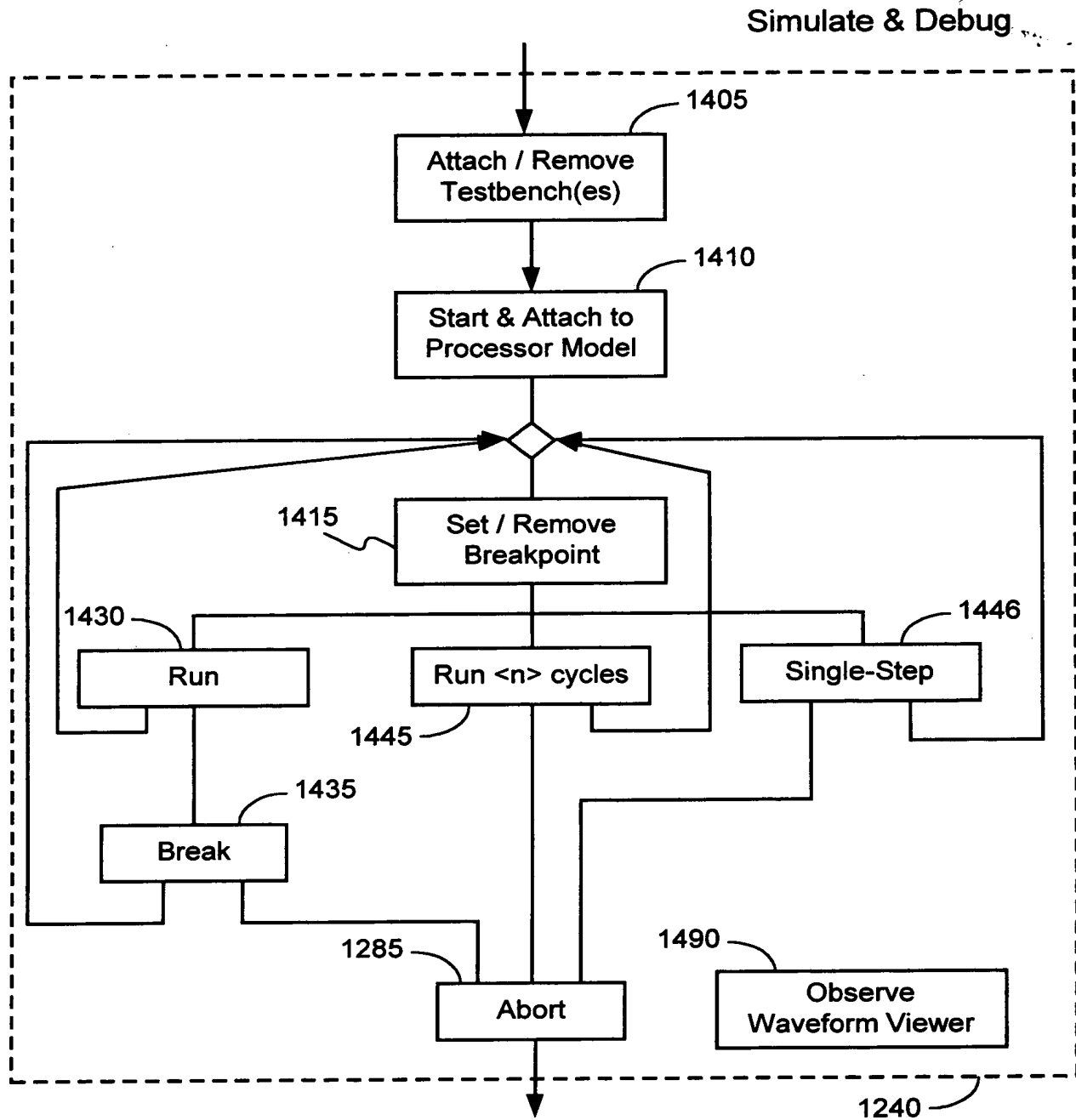


Figure 14

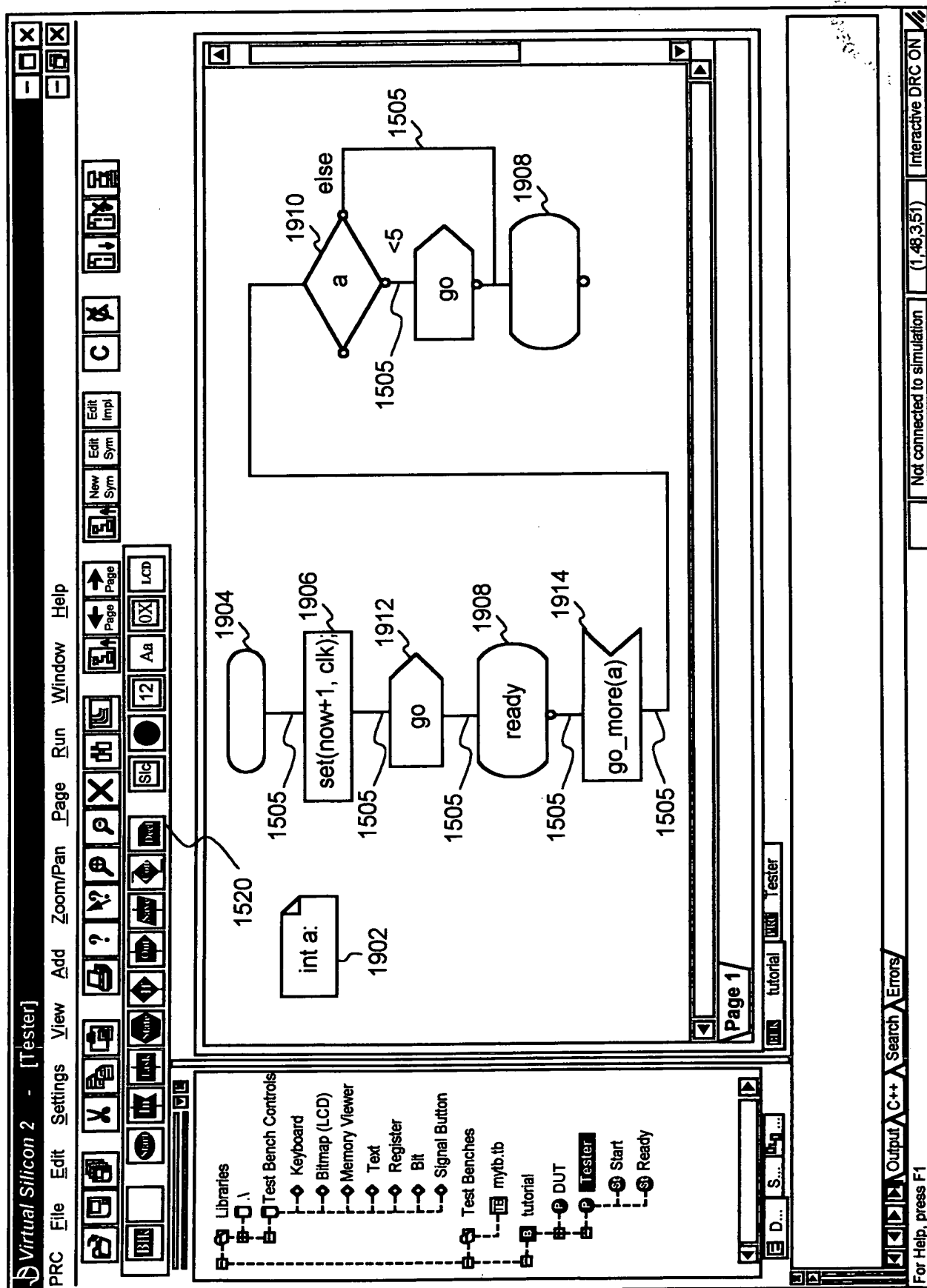


Figure 15

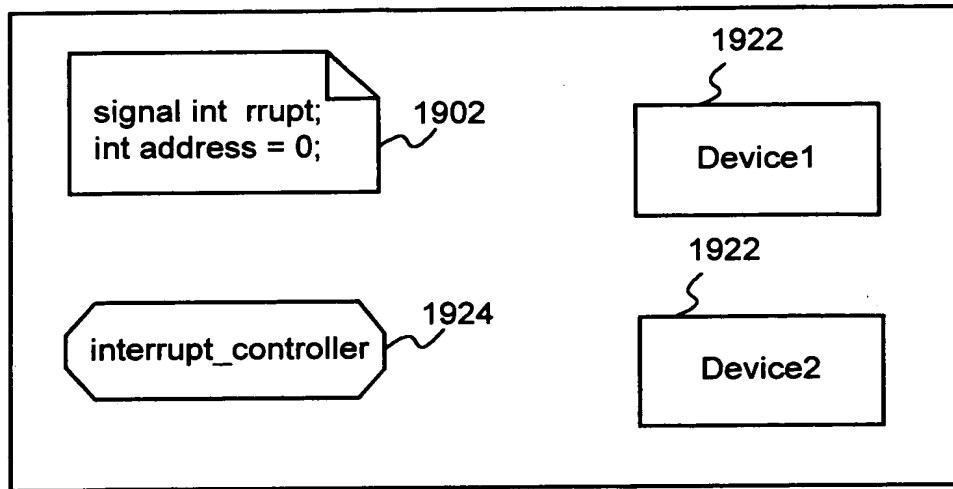


Figure 16

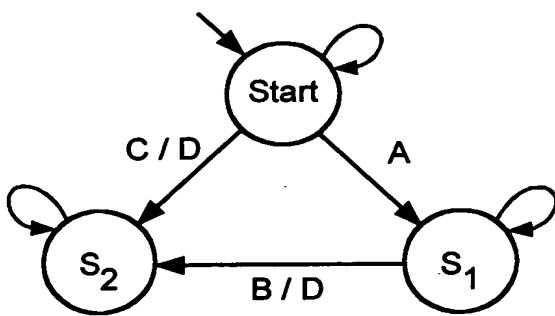


Figure 17(a)

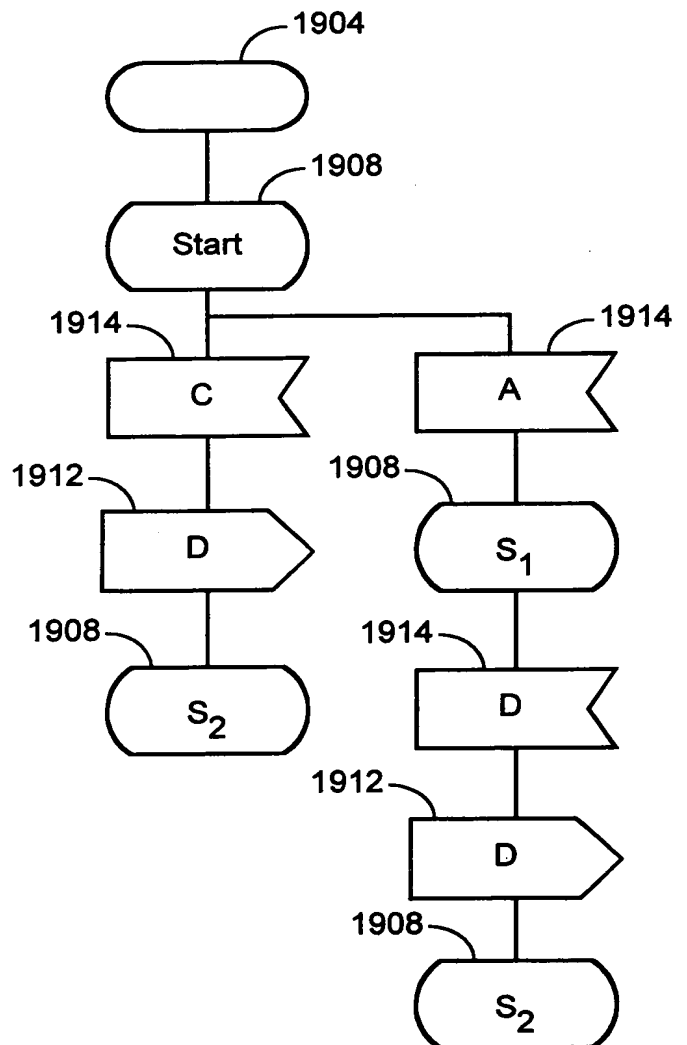
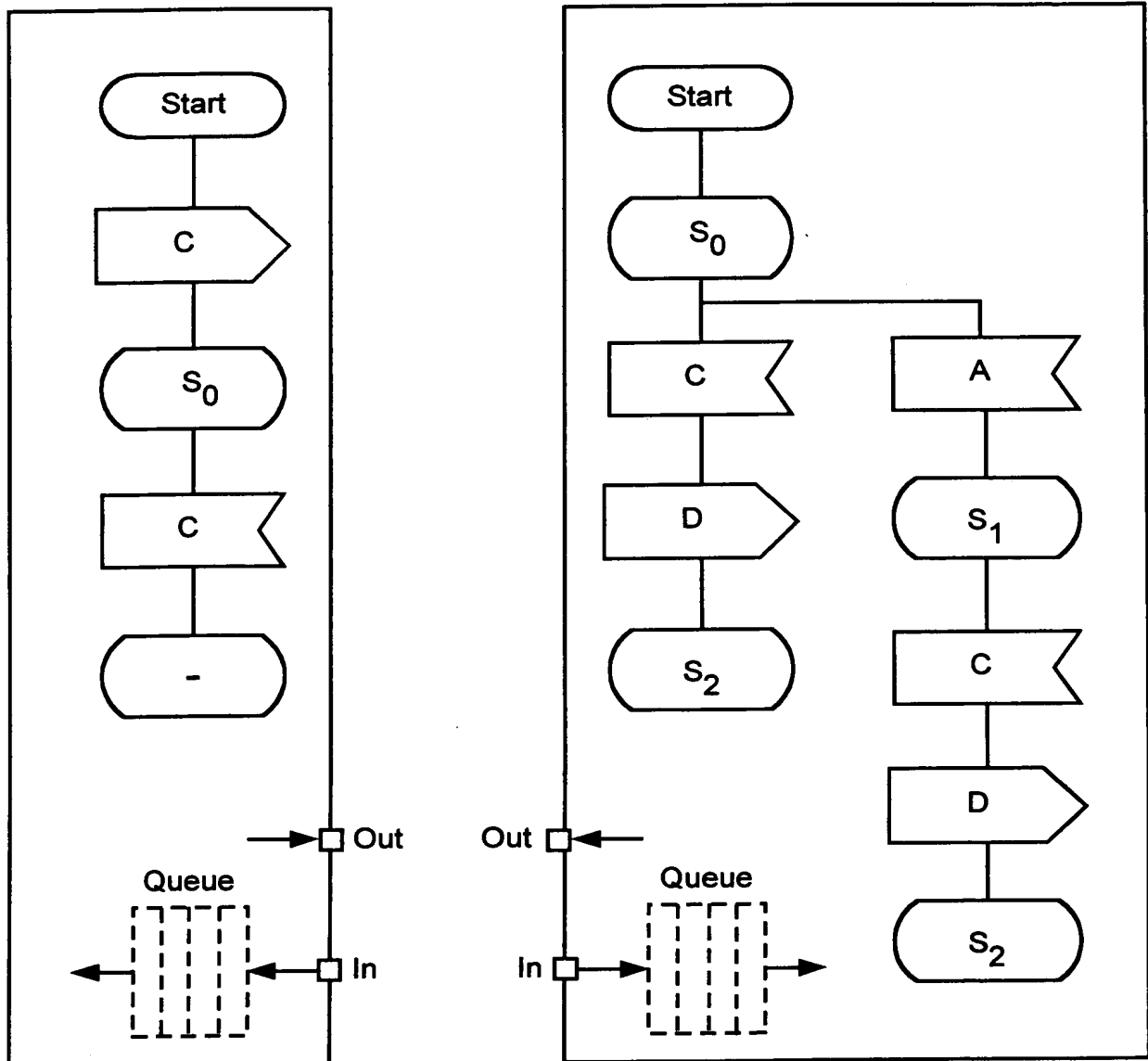


Figure 17(b)



Figur 18


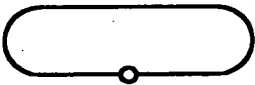
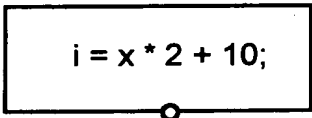

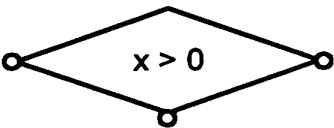
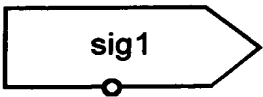
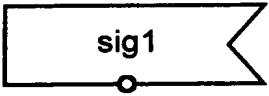
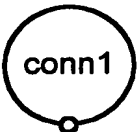
	Name	Graphical Symbol	De cription
1902	Declaration		defines local variables and signals.
1904	Start		Starting point of the Finite State Machine execution at initialization time
1906	Task		Execution block, containing ANSI-C statements to be executed
1908	State		Location where FSM waits in until a triggering signal is received.
1910	Decision		Directs execution flow based on the result of expression evaluation inside the decision construct.
1912	Signal-Out		Sending of a communication signal (with an optional payload)
1914	Signal-In		Receiving of a communication signal (with an optional payload)
1916	Connector		Allows to split designs over multiple pages, and connects the control flow between these diff rent pages.

Figure 19A

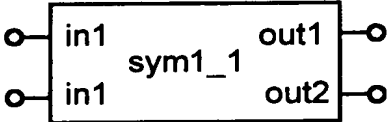


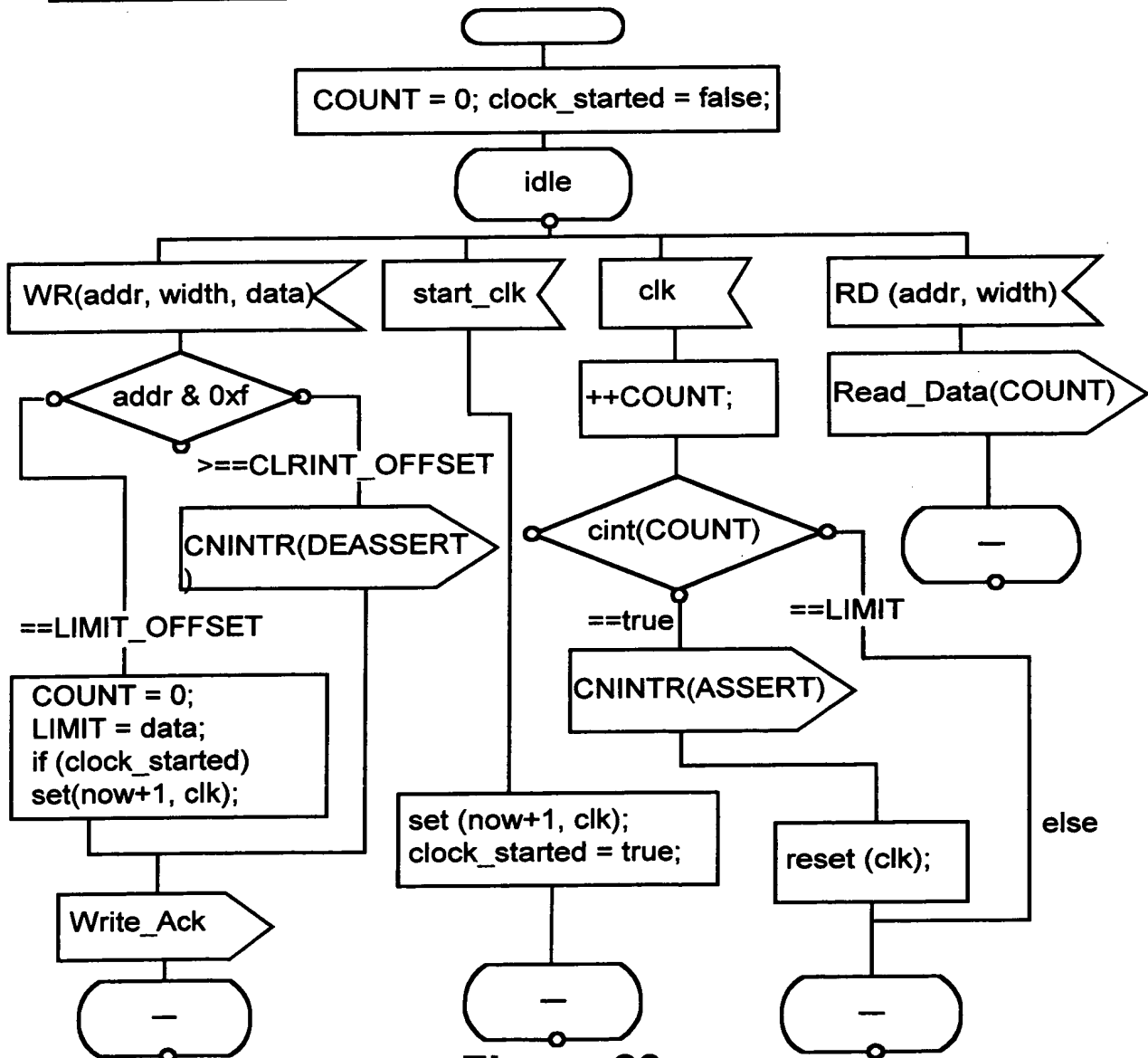
1920	Symbol		Captures design hierarchy and structure. Communication is done through pins on the outline of the symbol. Allows to re-use functional behavior by supporting multiple instances
1922	Blocks		Captures design hierarchy and structure. Communication is done through signals declared at higher scopes. Communication is done by signal name matching (rather than pin connection). A block can contain multiple processes.
1924	Process		Acts as leaf node in the design hierarchy, and captures a single FSM. By definition, all processes are concurrent at all times.

Figure 19B

```
// External interface
extern_signal WR(unsigned int, unsigned int,
unsigned int);
extern_signal RD(unsigned int, unsigned int);
extern_signal Write_Ack;
extern_signal Read_Data(unsigned int);
extern_signal CNTINTR(unsigned int);
// Local variables
signal start_clk;
clock clk;
bool clock_started;
unsigned int LIMIT; //write register
VS_int COUNT;
//temp vars
unsigned int data, width, addr;
```



Figur 20

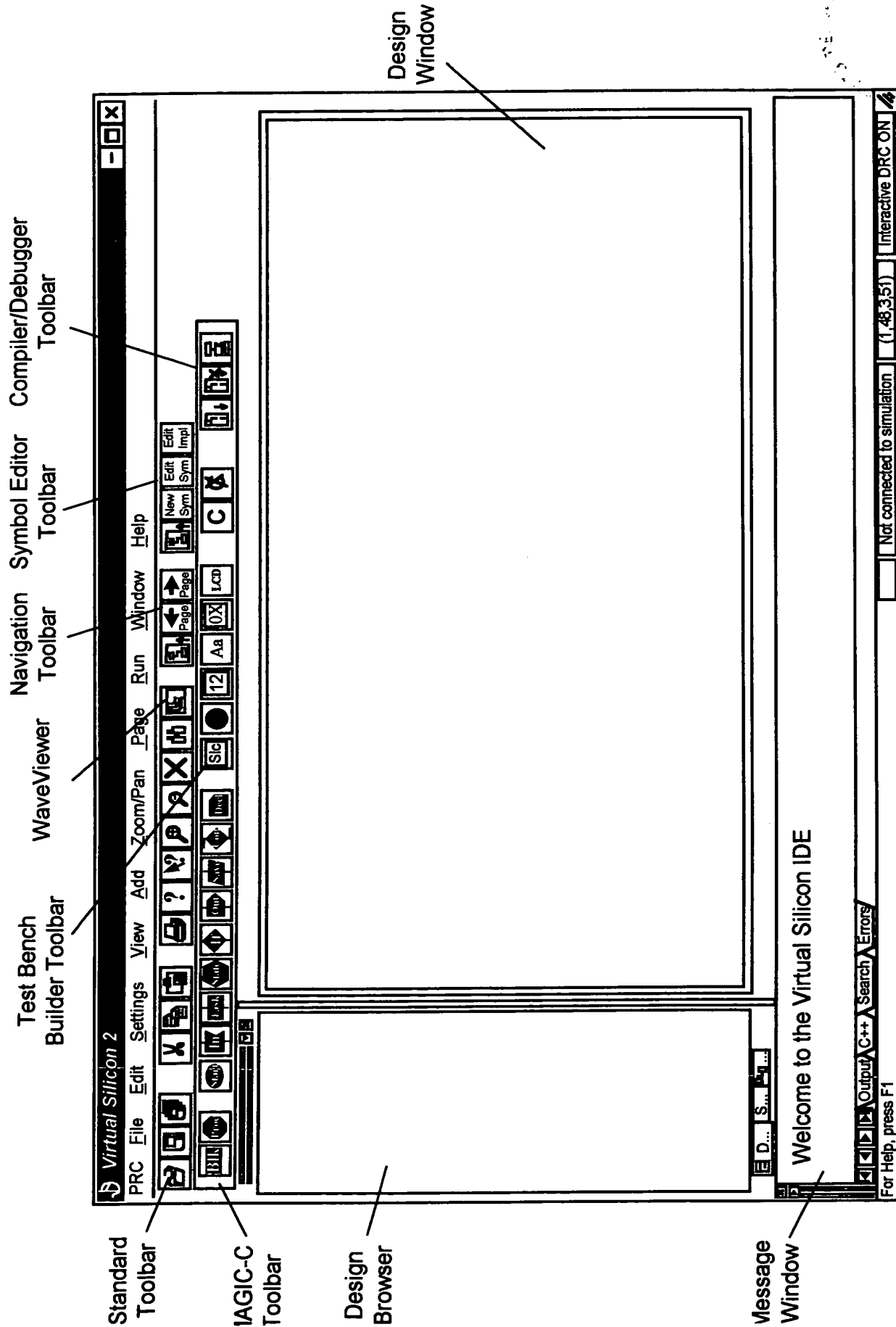
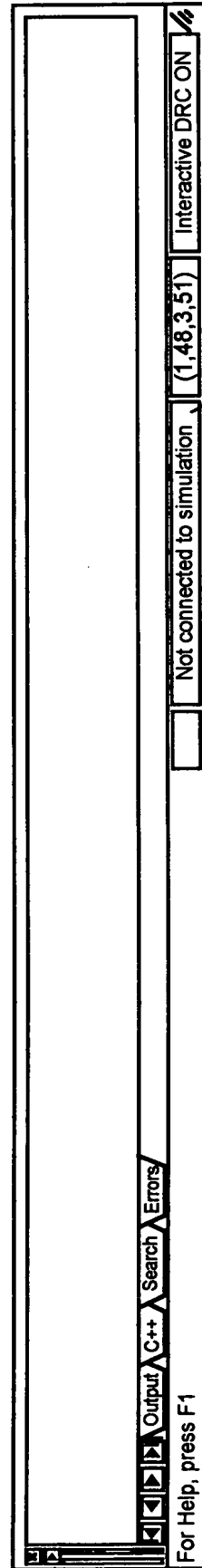


Figure 21



Standard Toolbar

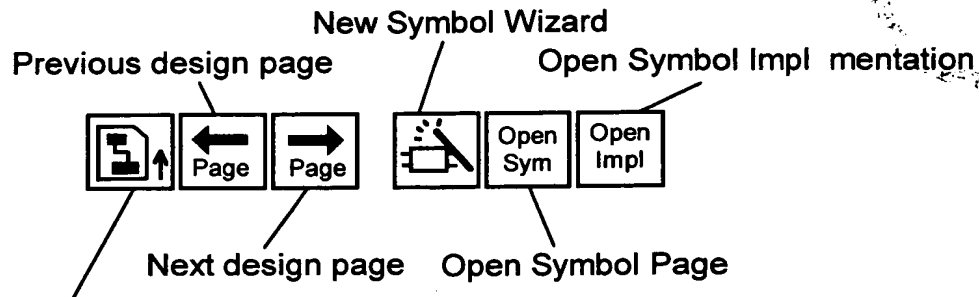
Figure 22



Status Bar

Message Window, Showing Different Message Tabs, and the Status Bar

Figure 23



Move up one level (in hierarchy)

Figure 23

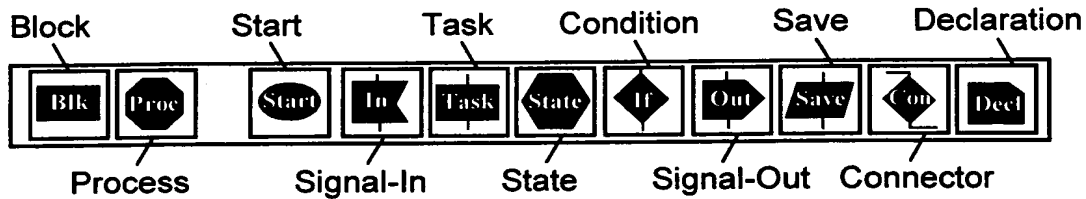


Figure 24

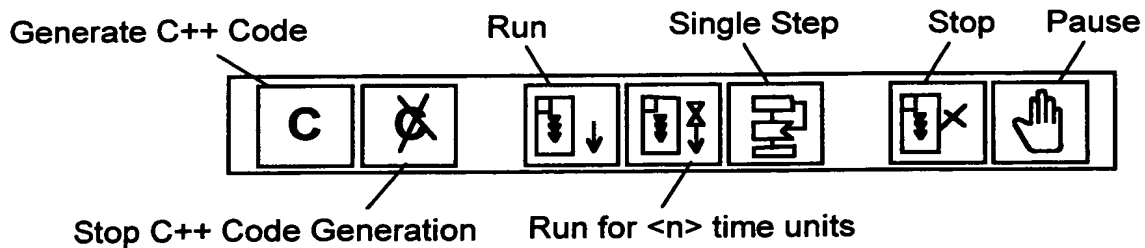


Figure 25

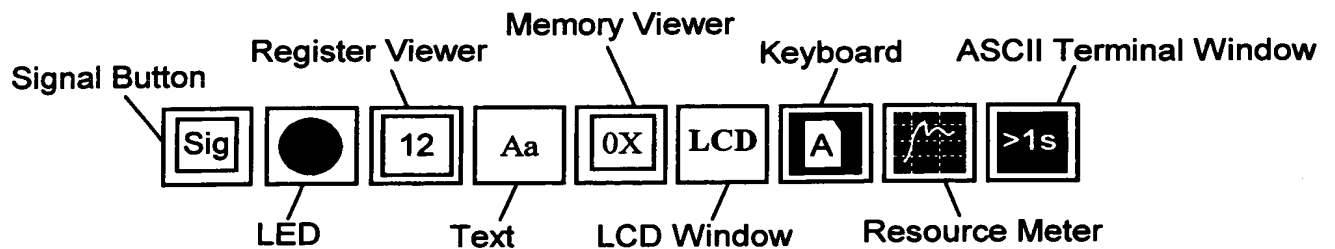


Figure 26

Inventor(s): Stephen L. Bade et al.
Atty. Docket No.: 22178-05012
Sheet 25 of 65

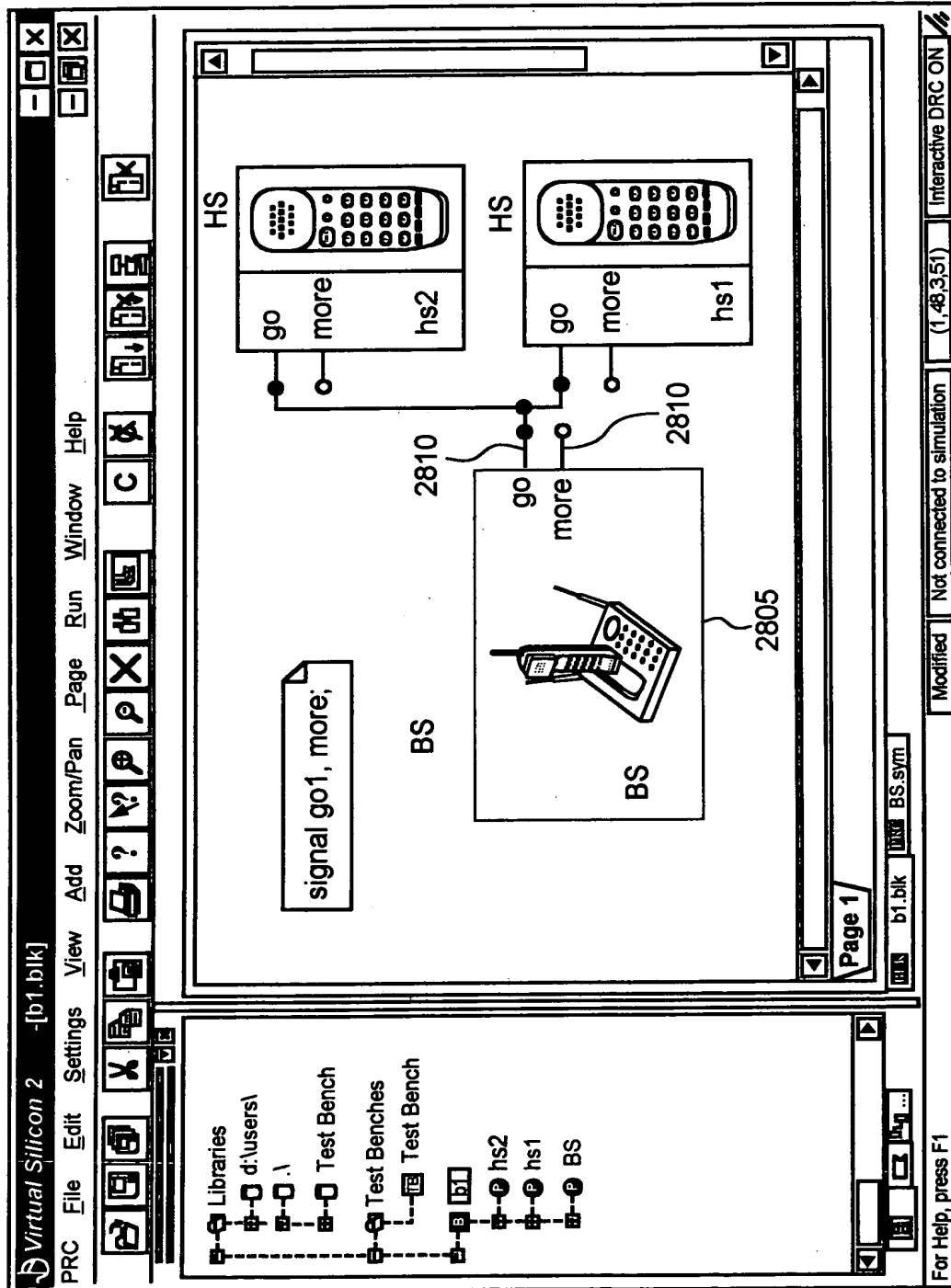


Figure 28

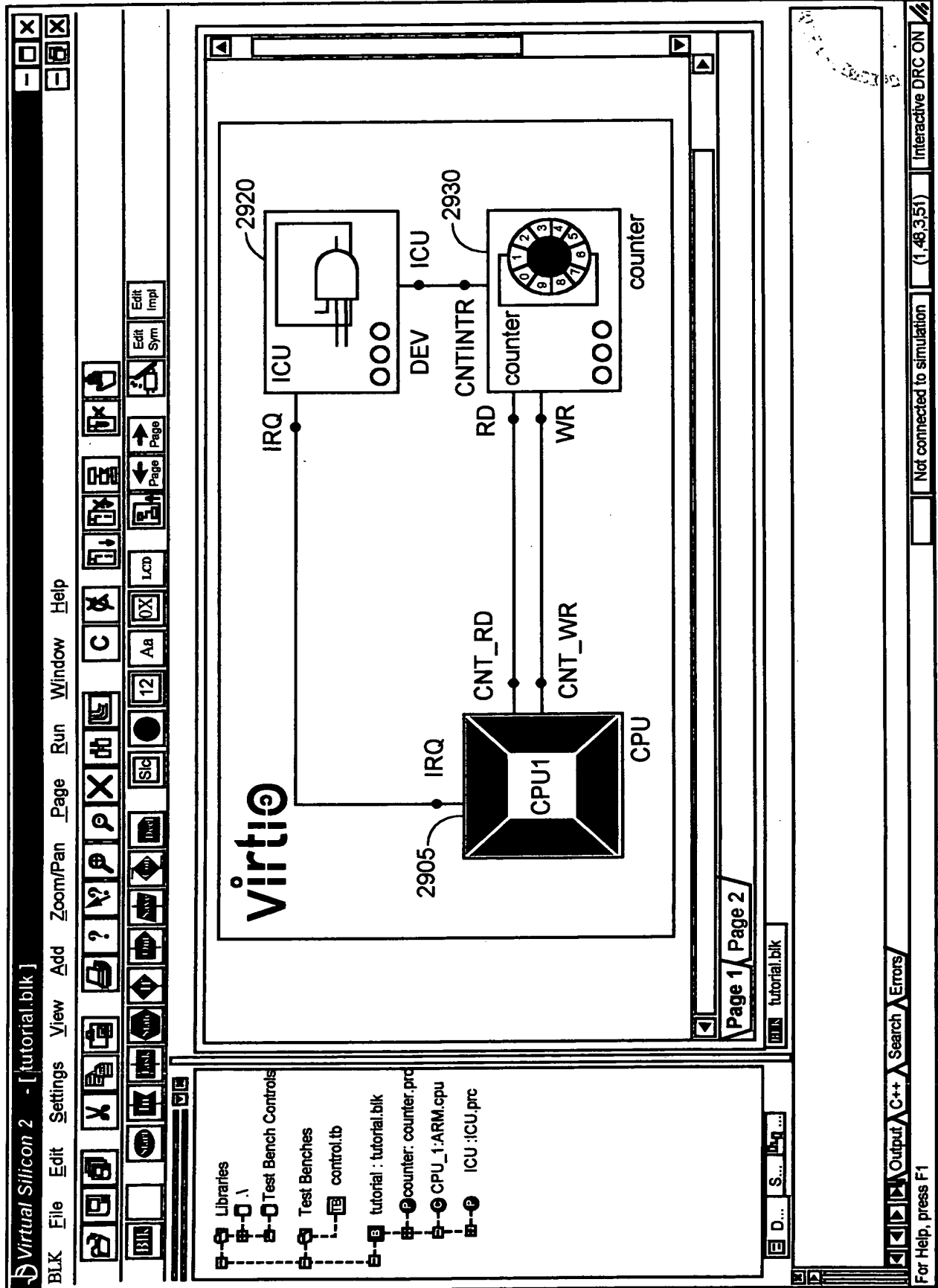


Figure 29

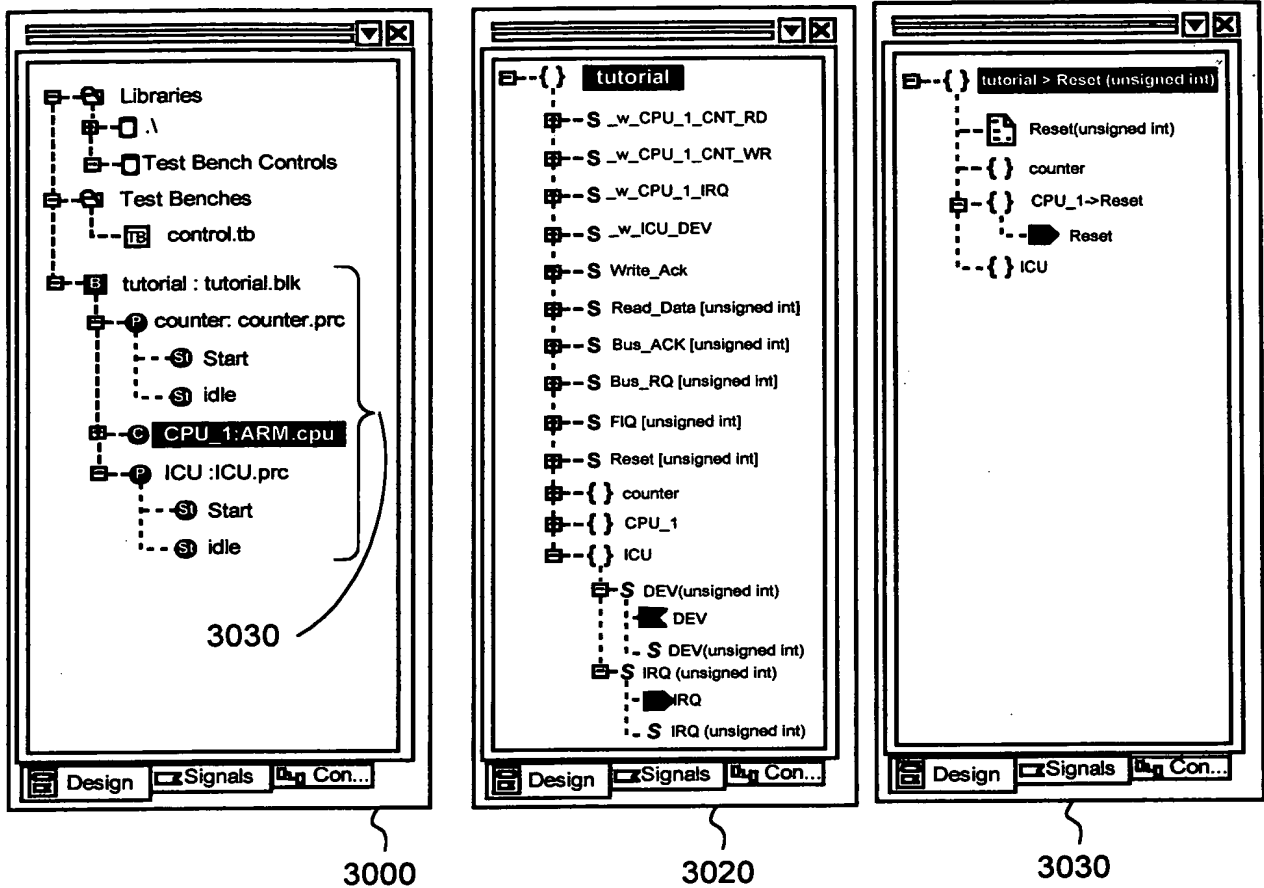
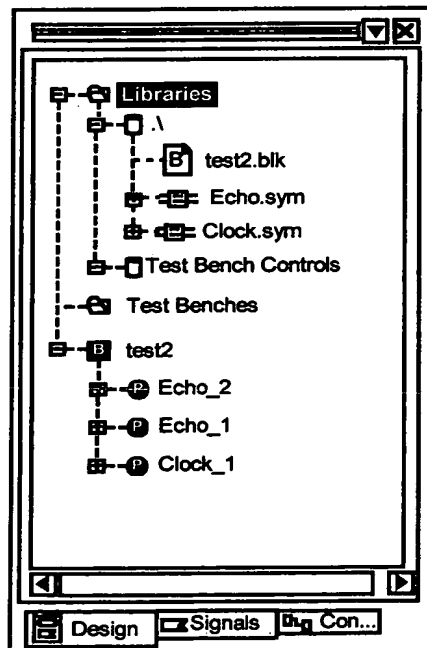


Figure 30



Figur 31

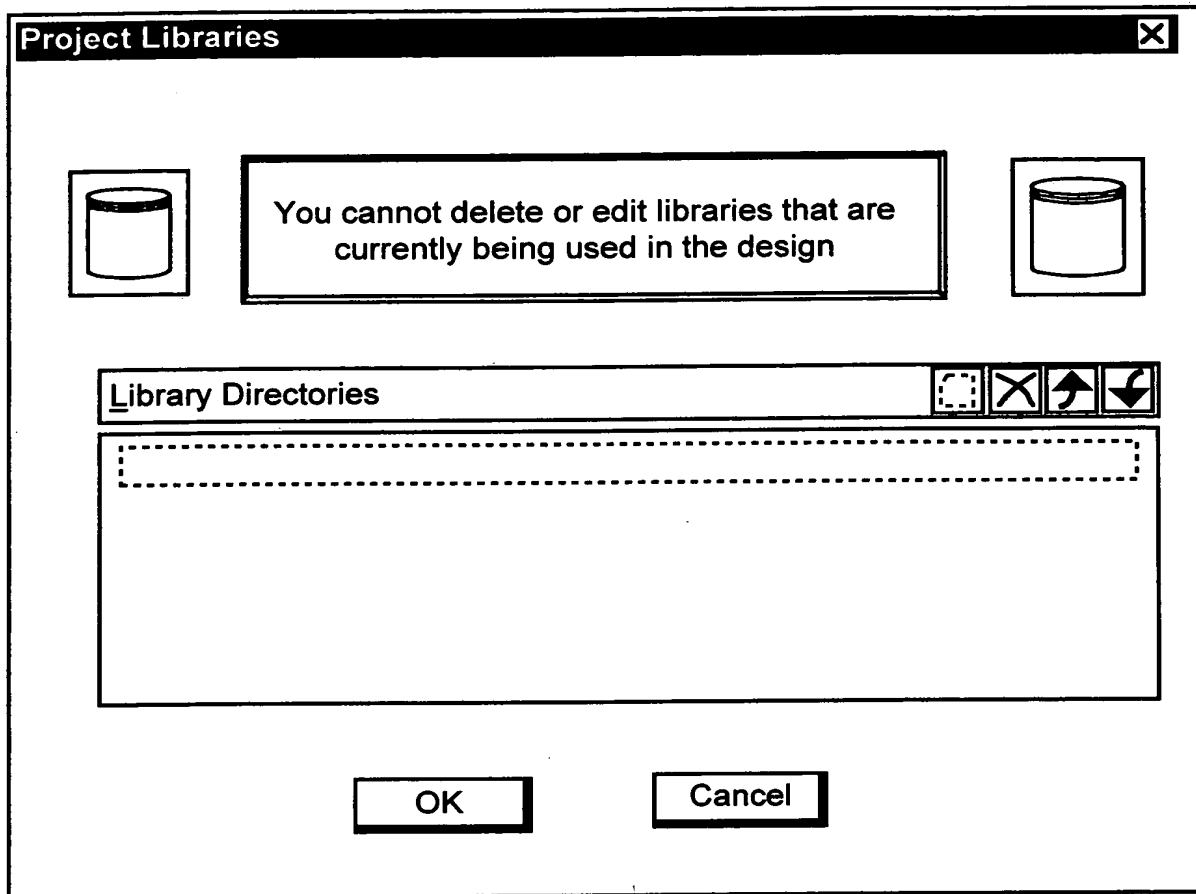


Figure 32

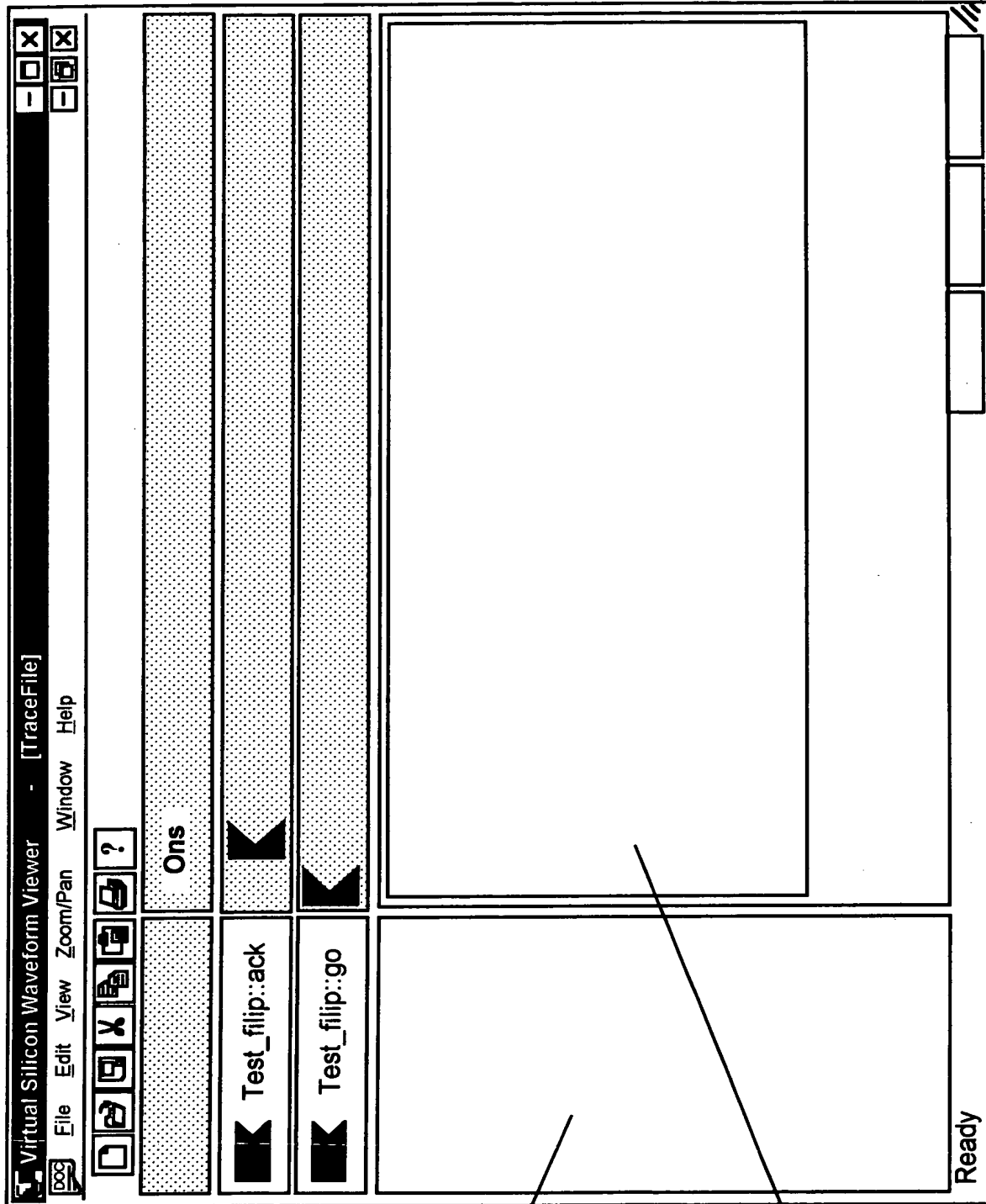


Figure 33A

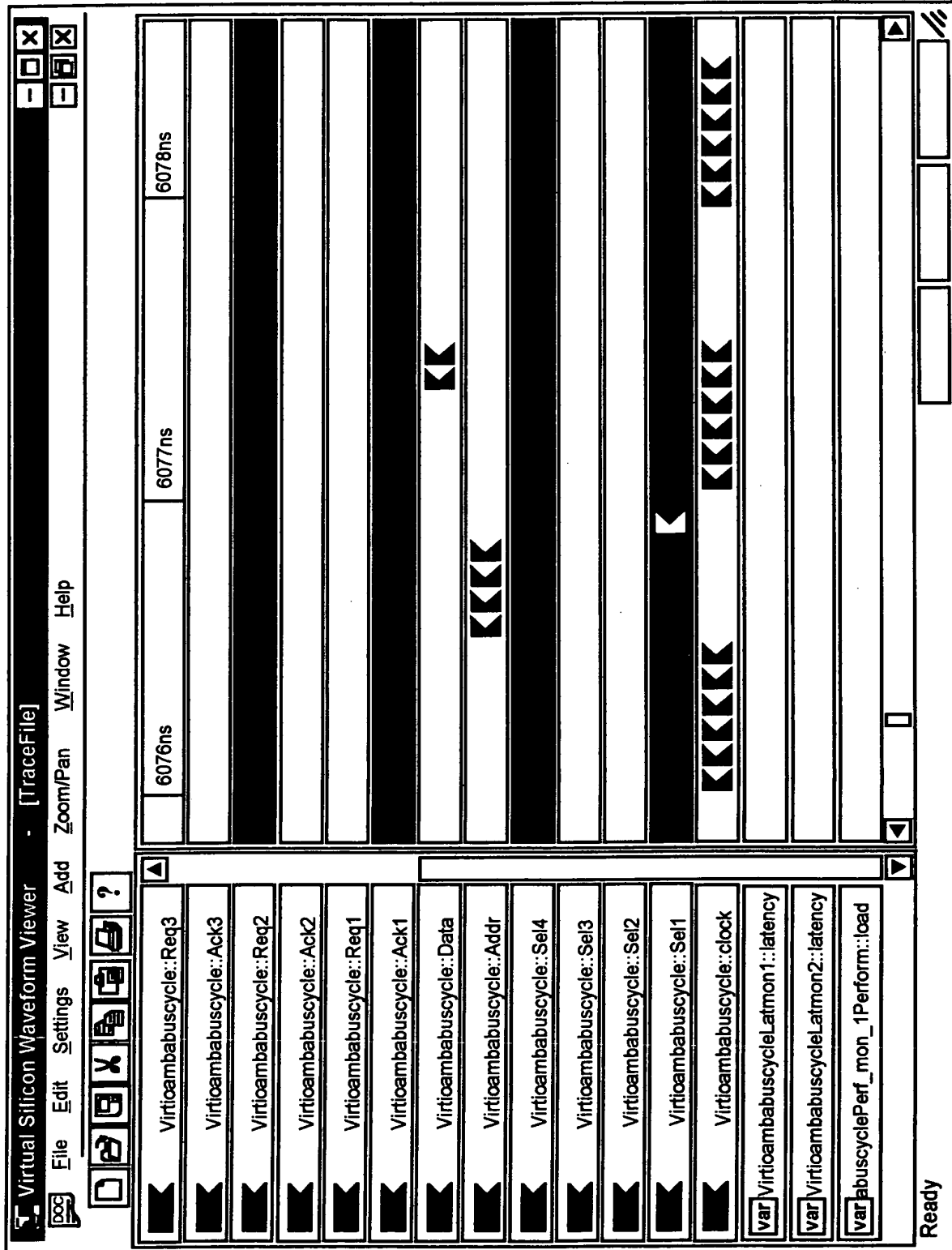


Figure 33B

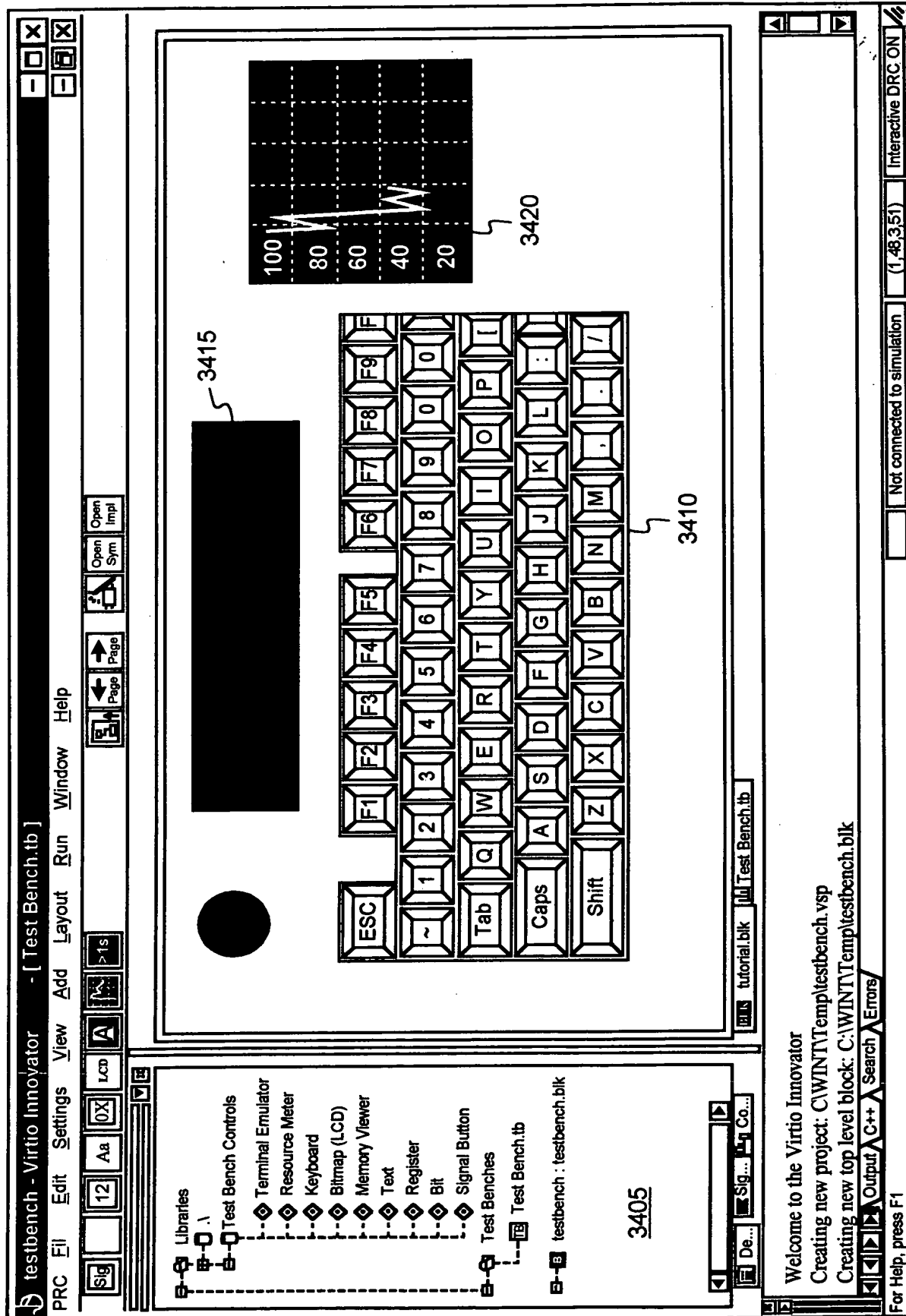


Figure 34A

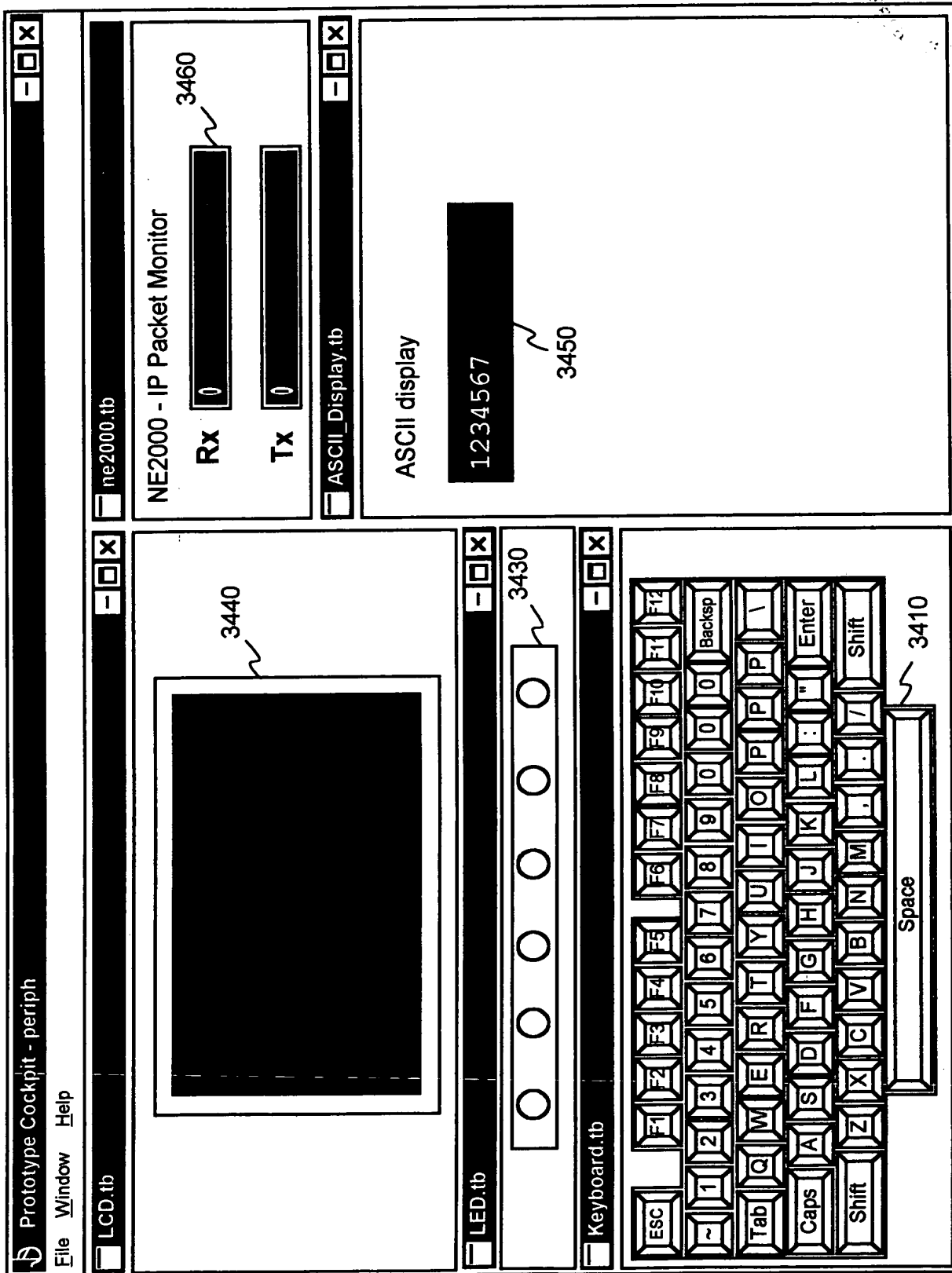
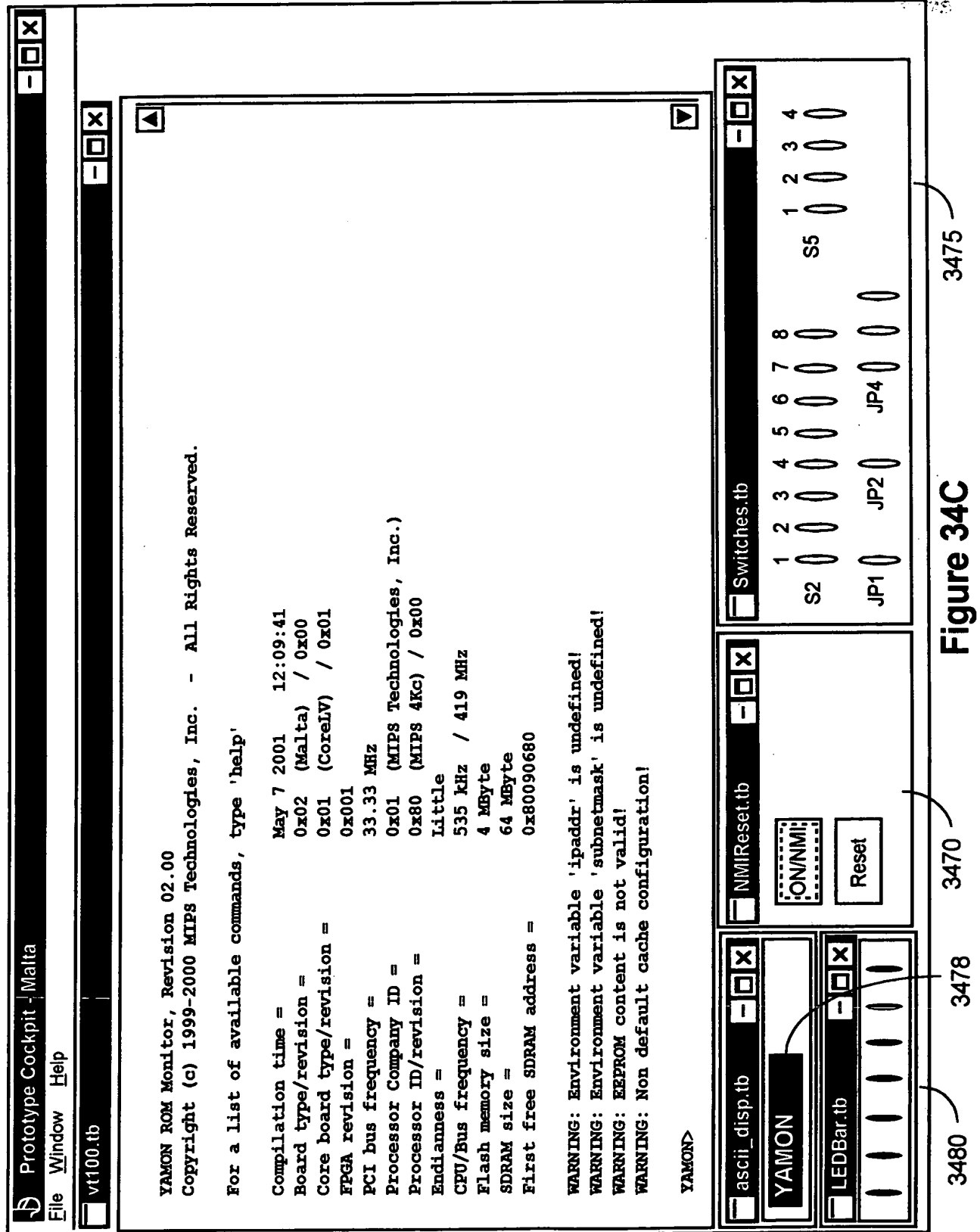


Figure 34B



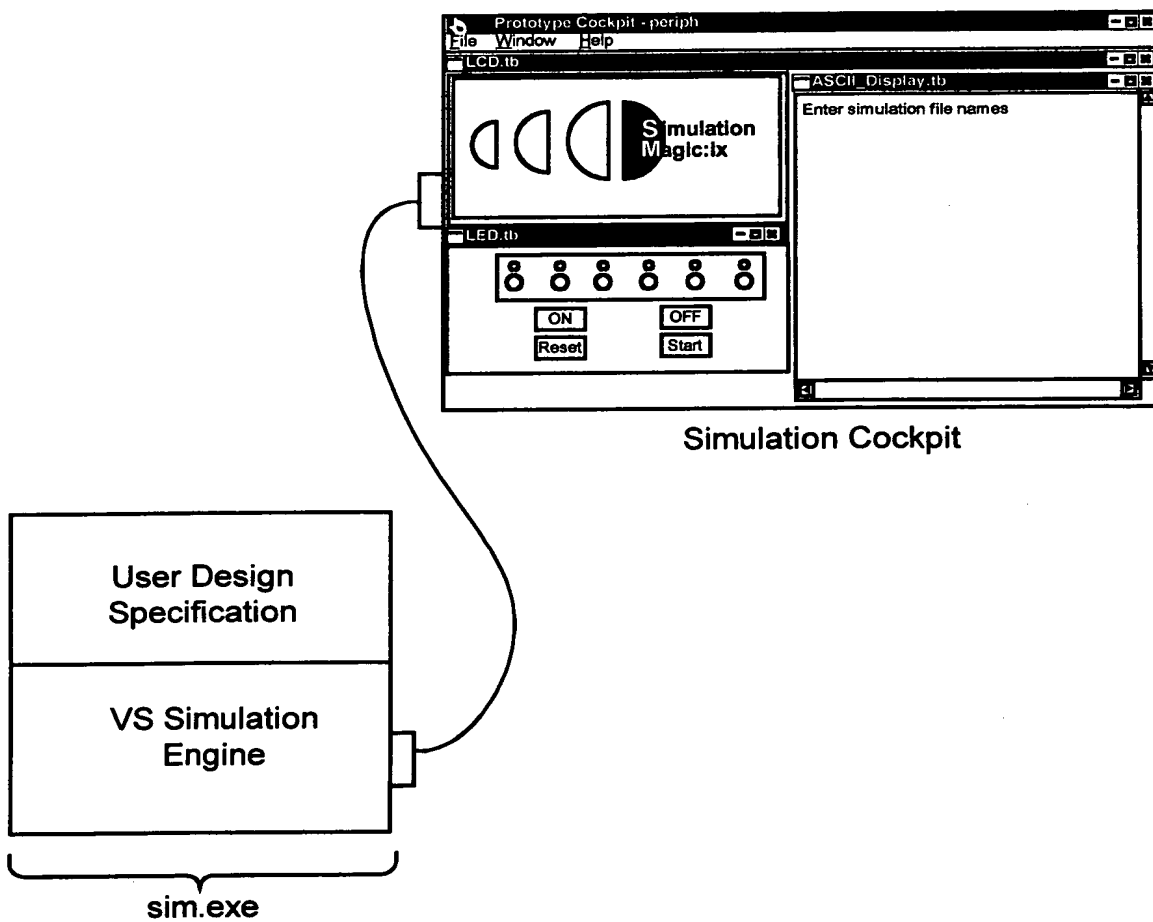


Figure 35

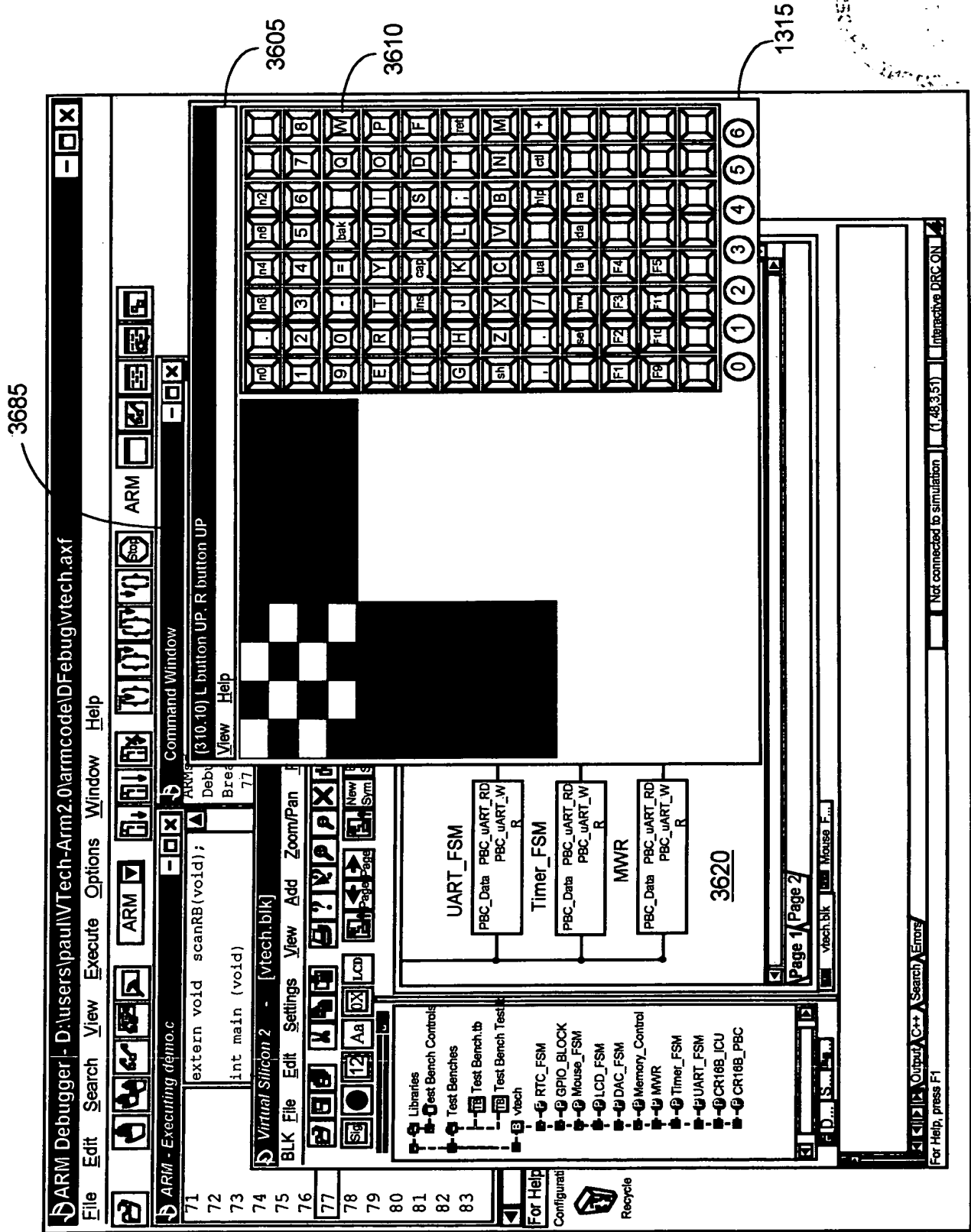


Figure 36A

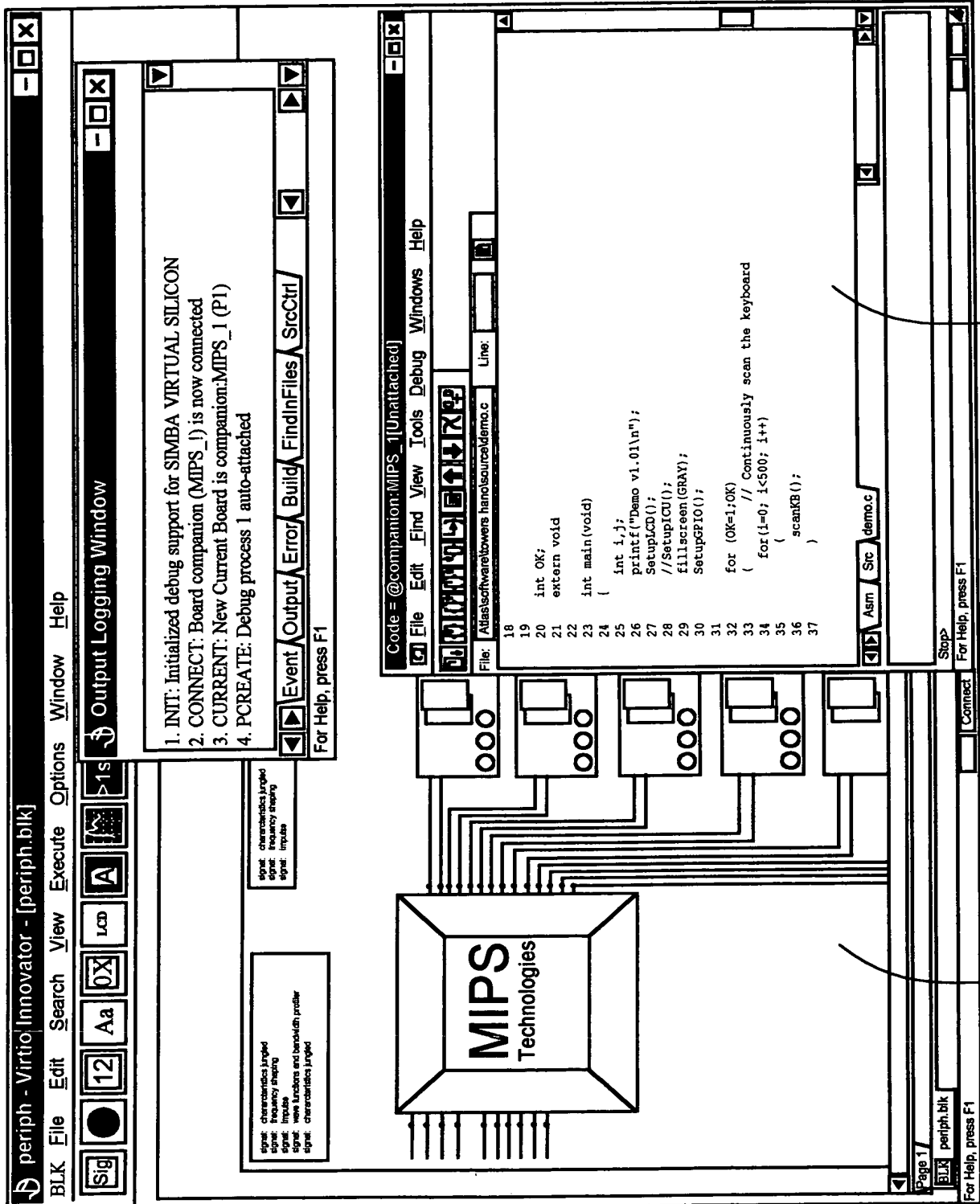


Figure 36B

3620

3685

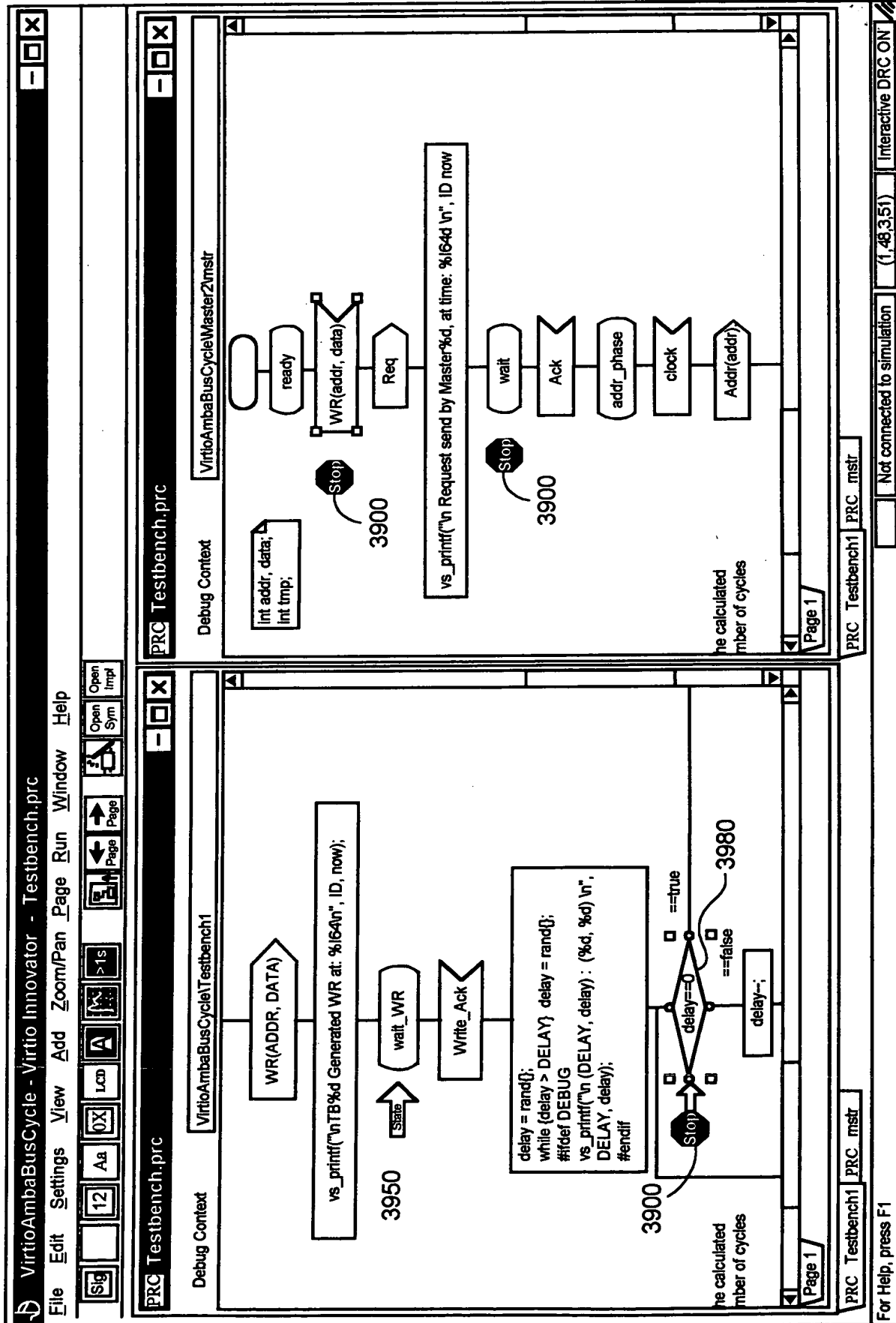


Figure 36C

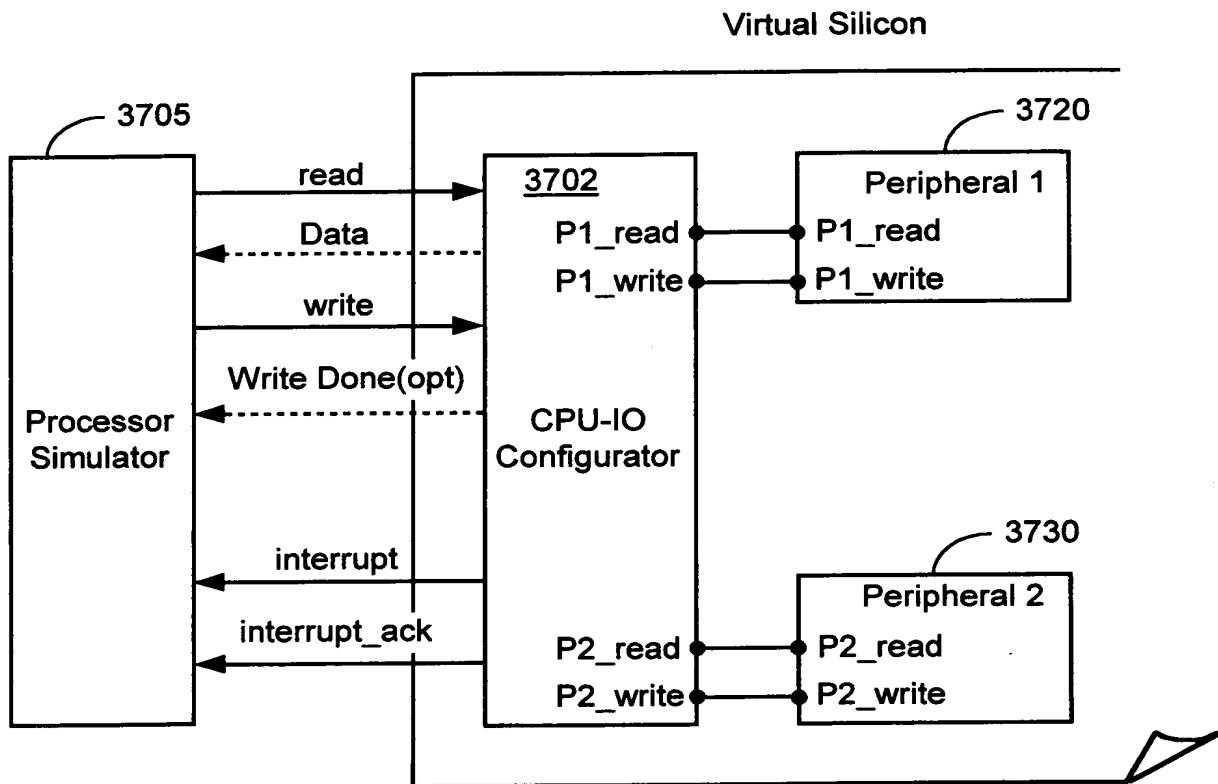


Figure 37

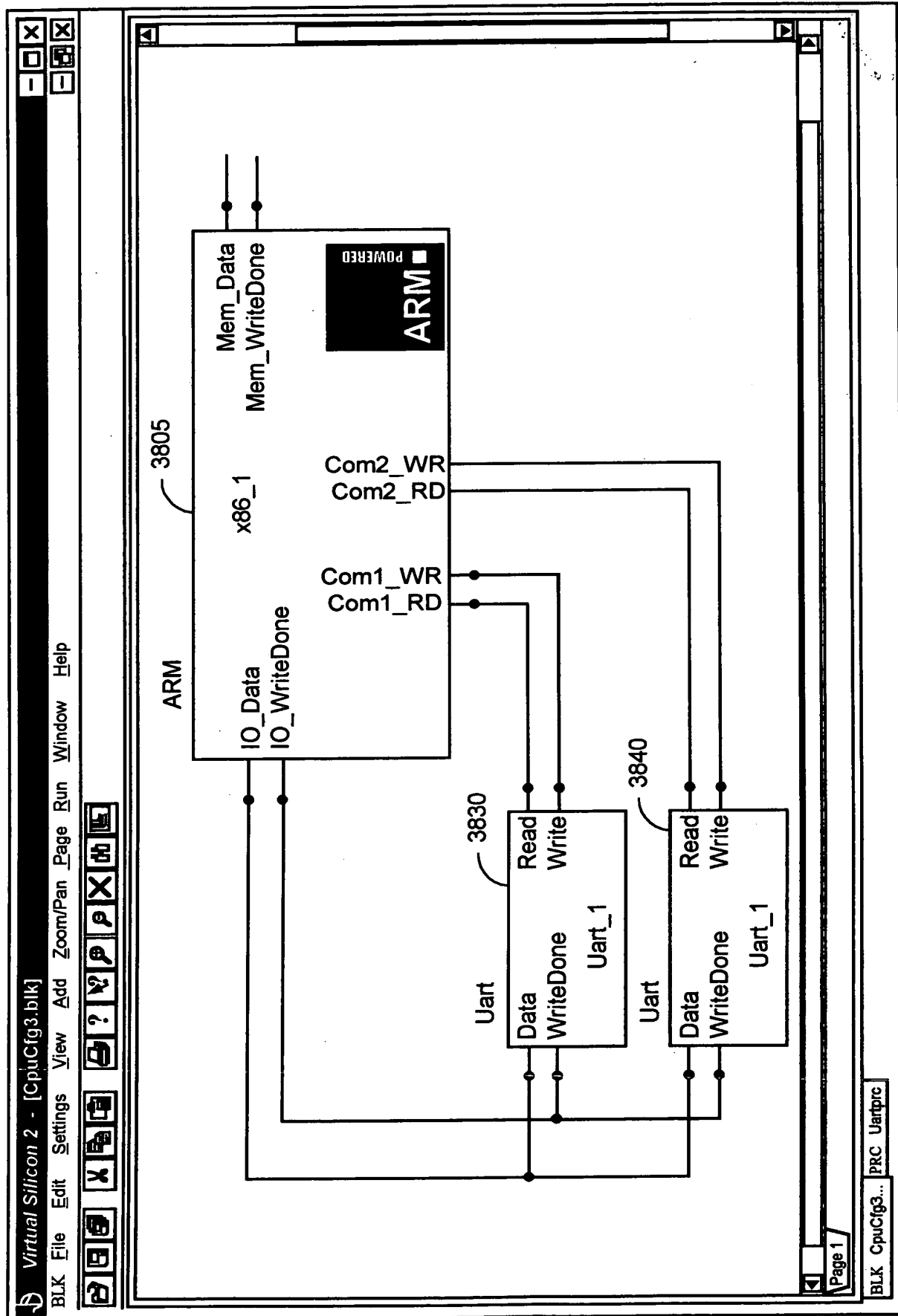


Figure 38

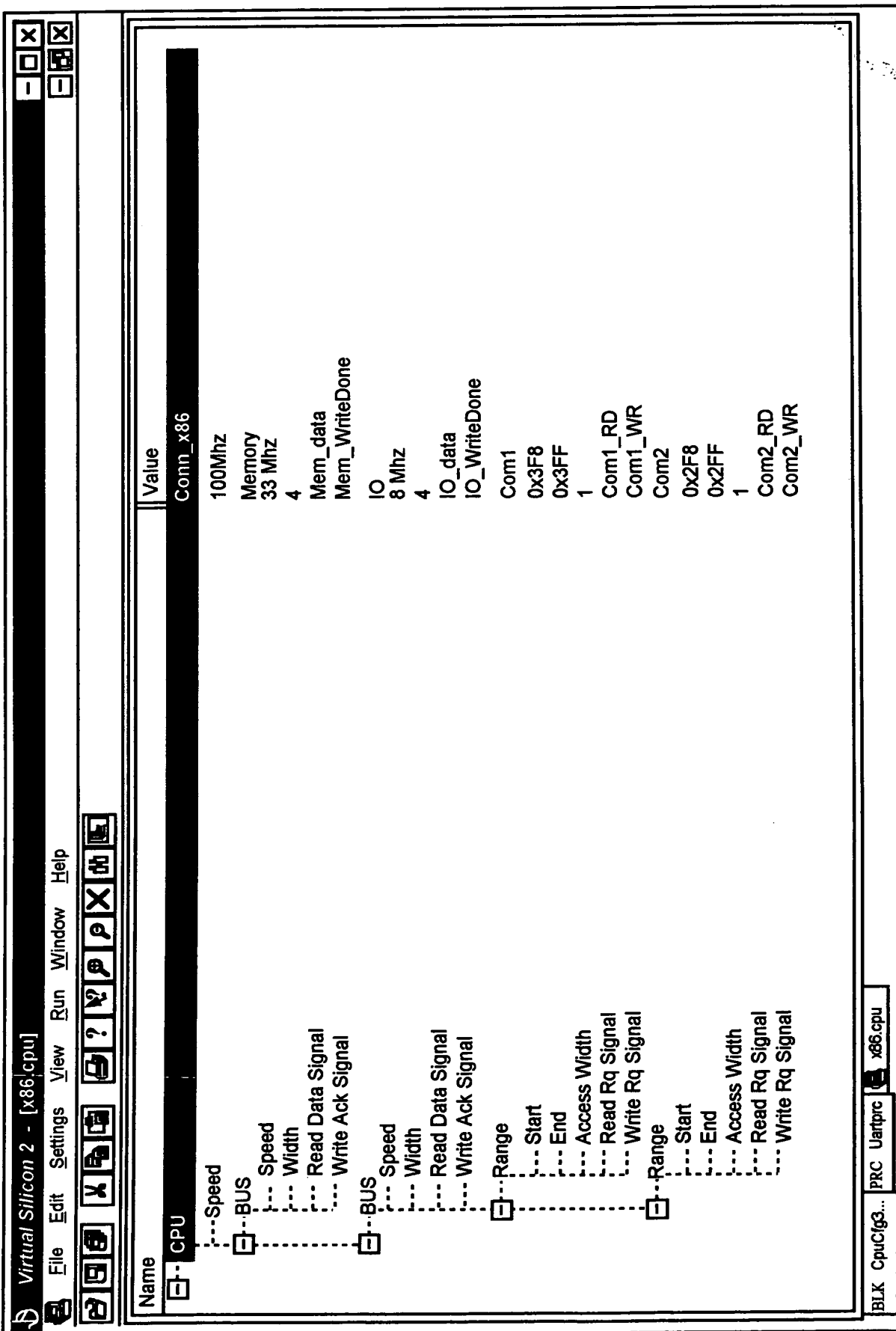


Figure 39

Name	Value
<input type="checkbox"/> CPU	AmCPU
Speed	100Mhz
<input type="checkbox"/> Events: CPU -> Simulation	
Reset Signal	Rest
BUS_ACK Signal	Bus_ack
<input type="checkbox"/> Events: Simulation -> CPU	
BUS_RQ	Bus_RQ
<input type="checkbox"/> Interrupt Support	Yes
FIRQ Signal	FIQ
IRQ Signal	IRQ
<input type="checkbox"/> BUS	Memory
Speed	100Mhz
Width	4
Read Data Signal	Read_data
<input type="checkbox"/> Write Timing	Variable
Write Ack Signal	Write_Ack
<input type="checkbox"/> Range	Counter
Type	Slave
Start	CNT_START_ADDR
End	CNT_END_ADDR
Access Width	4
Read Rq Signal	CNT_RD
Write Rq Signal	CNT_WR

Figure 40

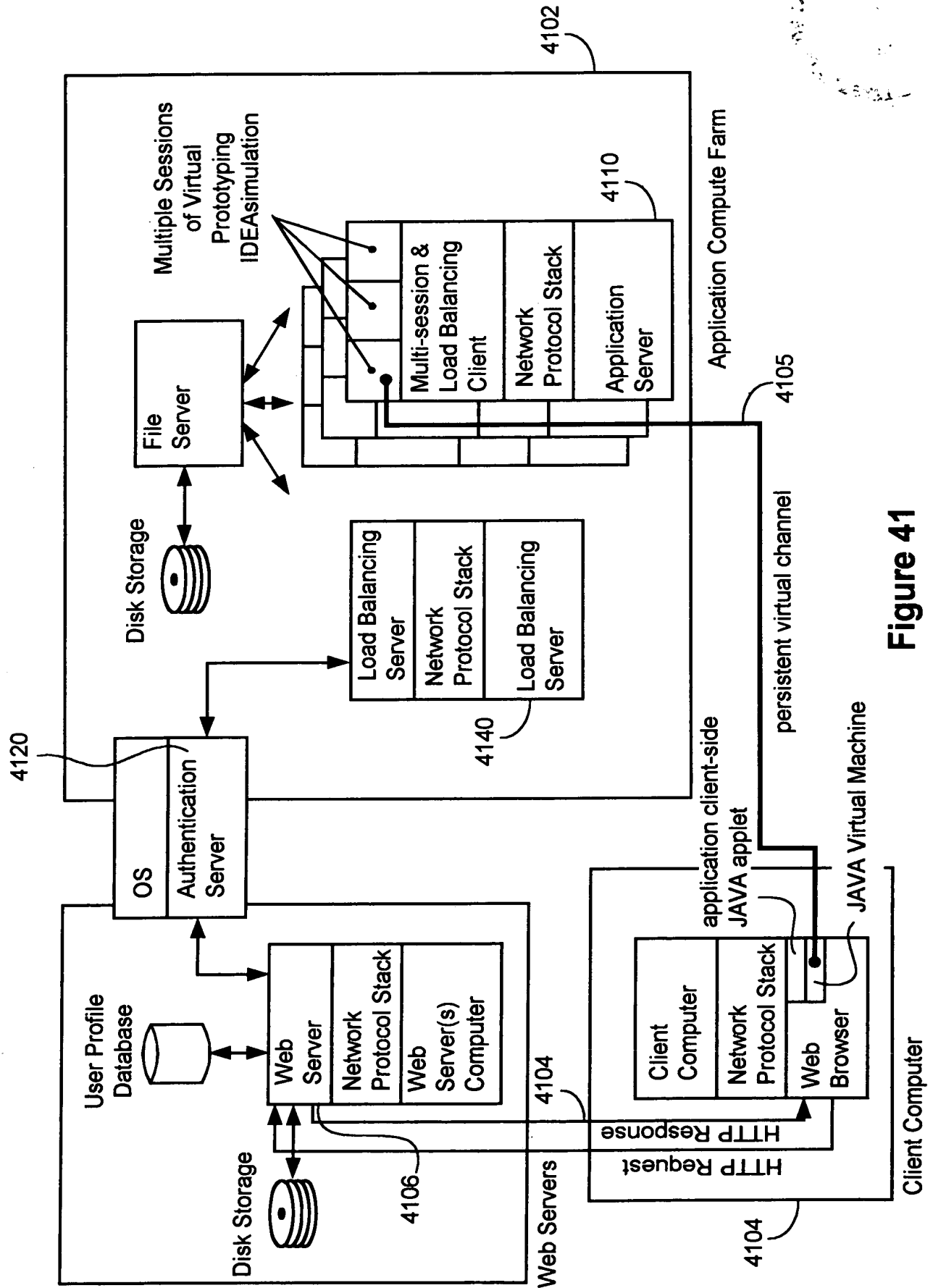
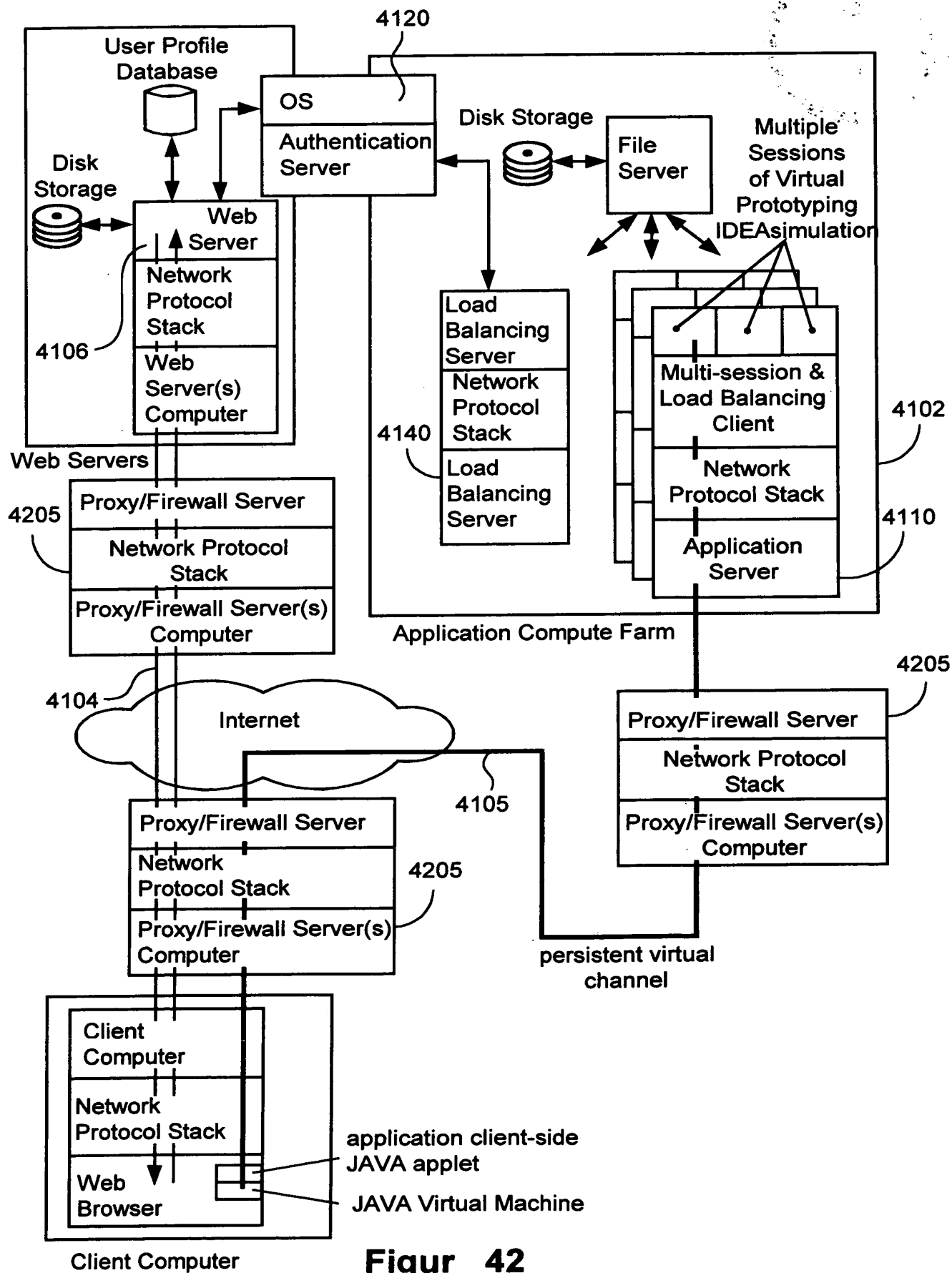


Figure 41



Figur 42

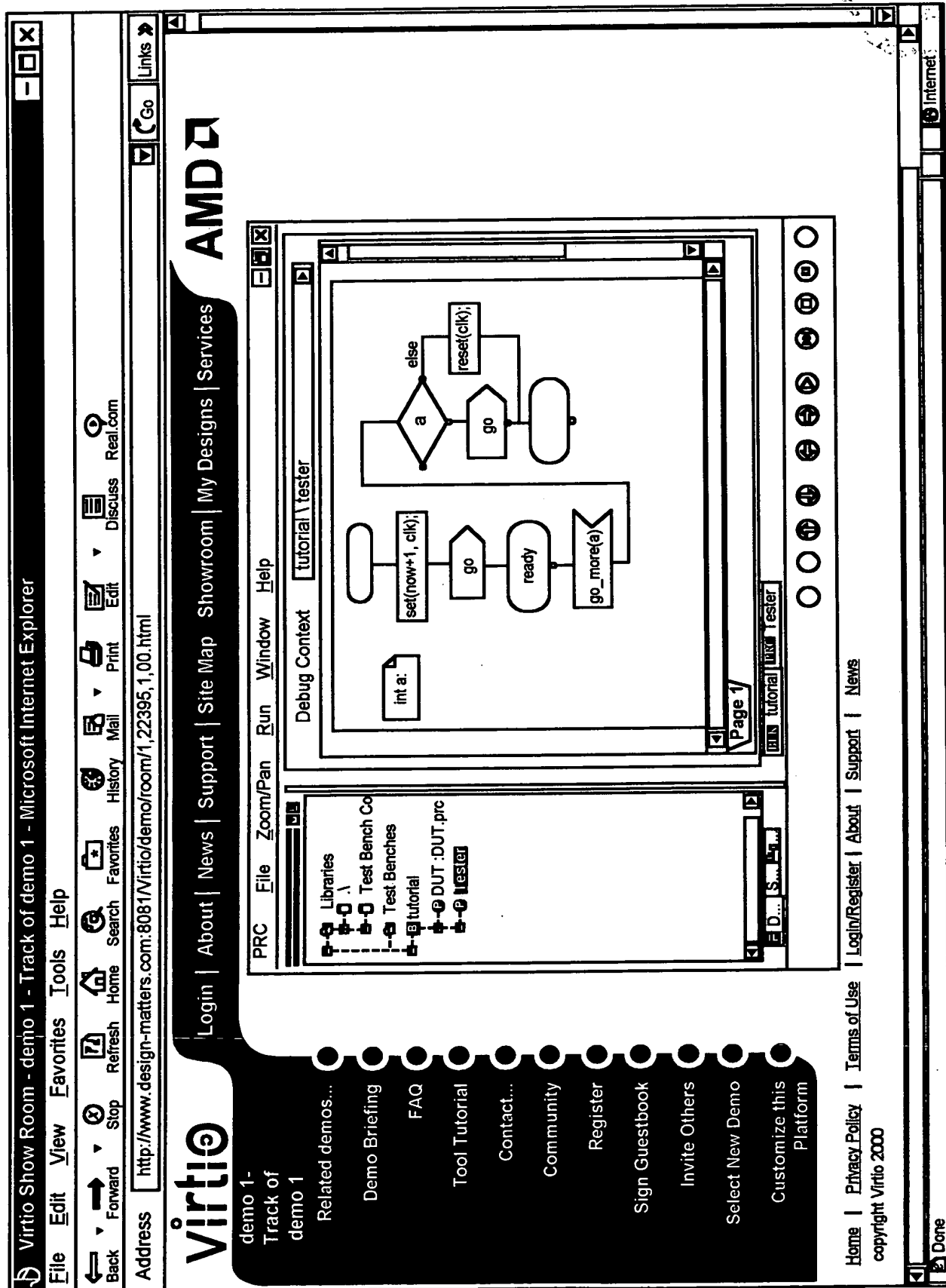


Figure 43

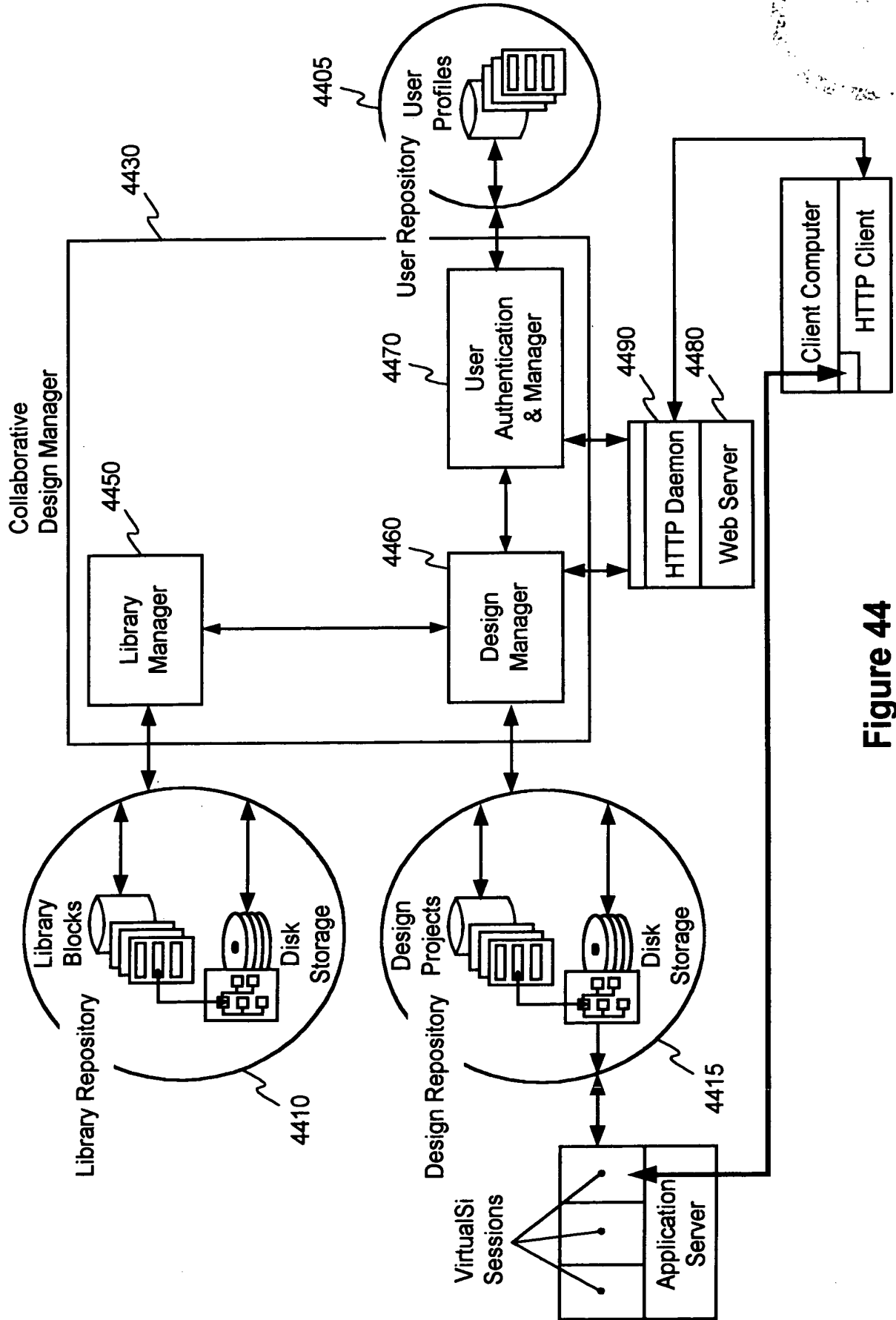




Figure 44



 Clock_1	Scope Name
 clk(int)	Signal name and declaration







 t	Timer or clock name and declaration
S clk	Local name of a signal coming from the upper scope (inherited)
 myClk 	Timer or clock is being set
	Signal is being sent
	Signal is being received
	Signal is being saved

Figure 45

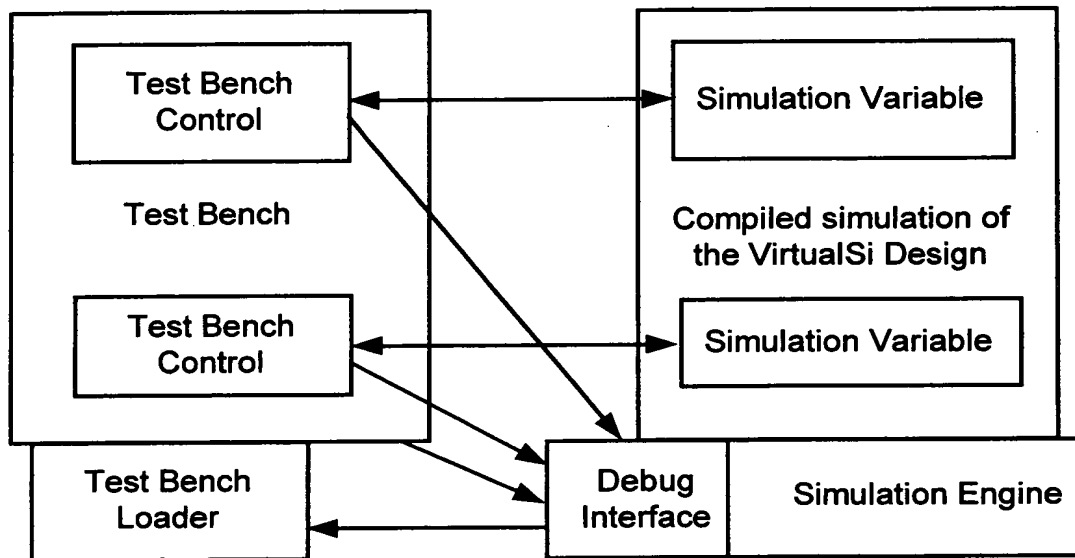


Figure 46

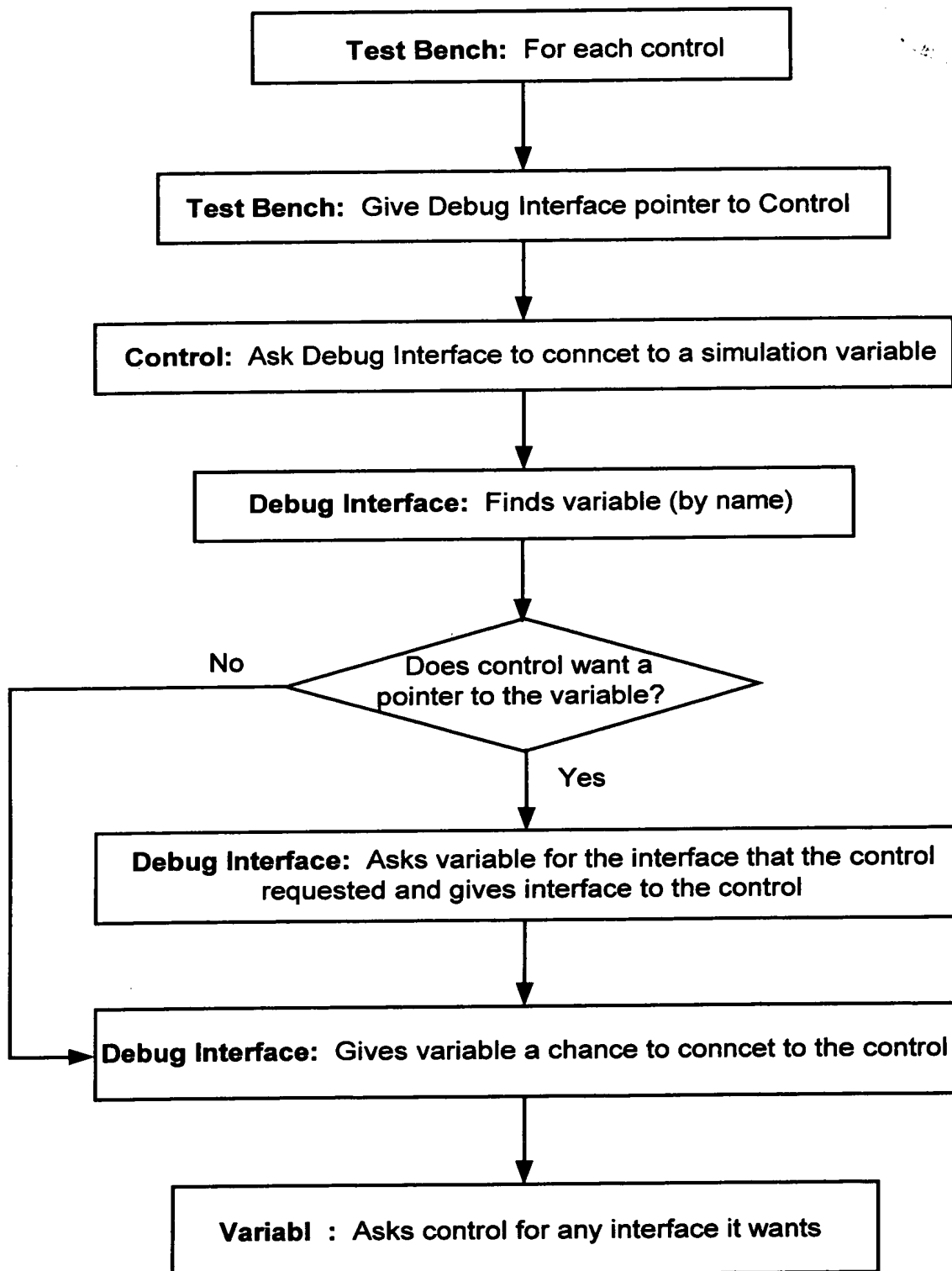


Figure 47

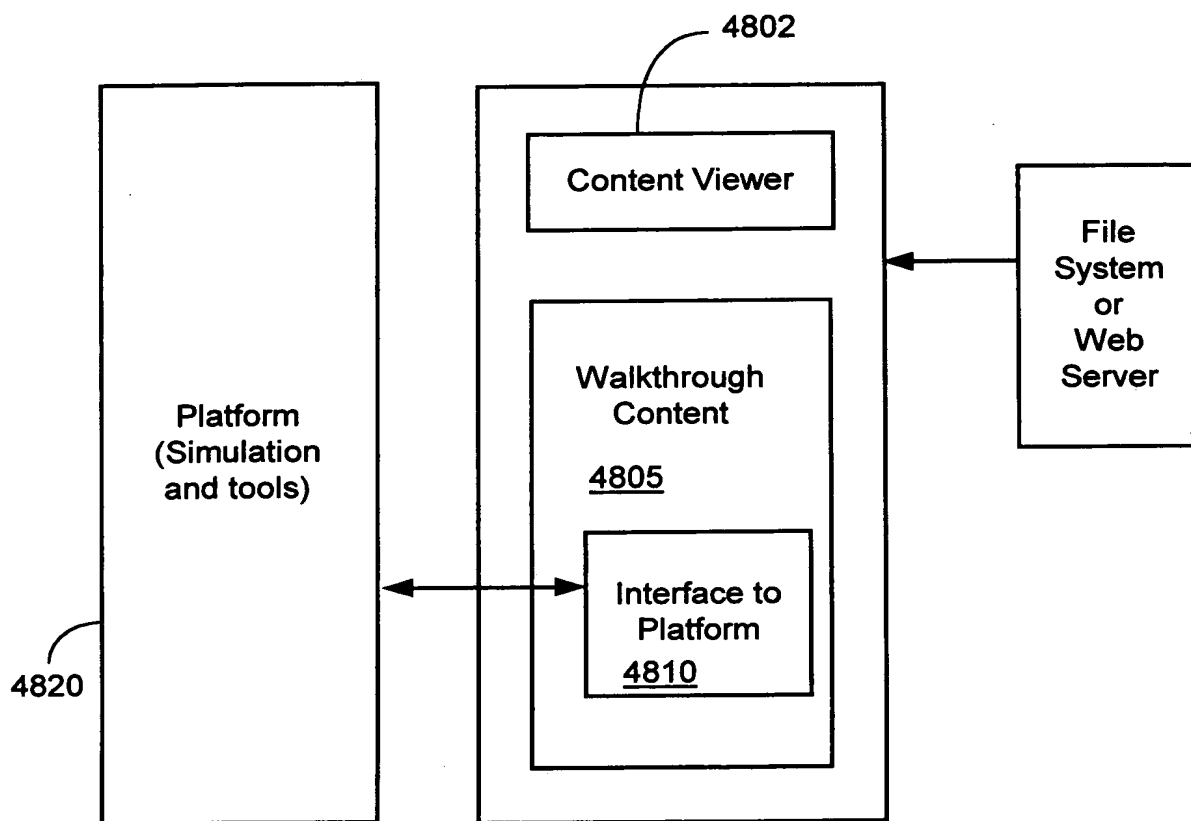


Figure 48

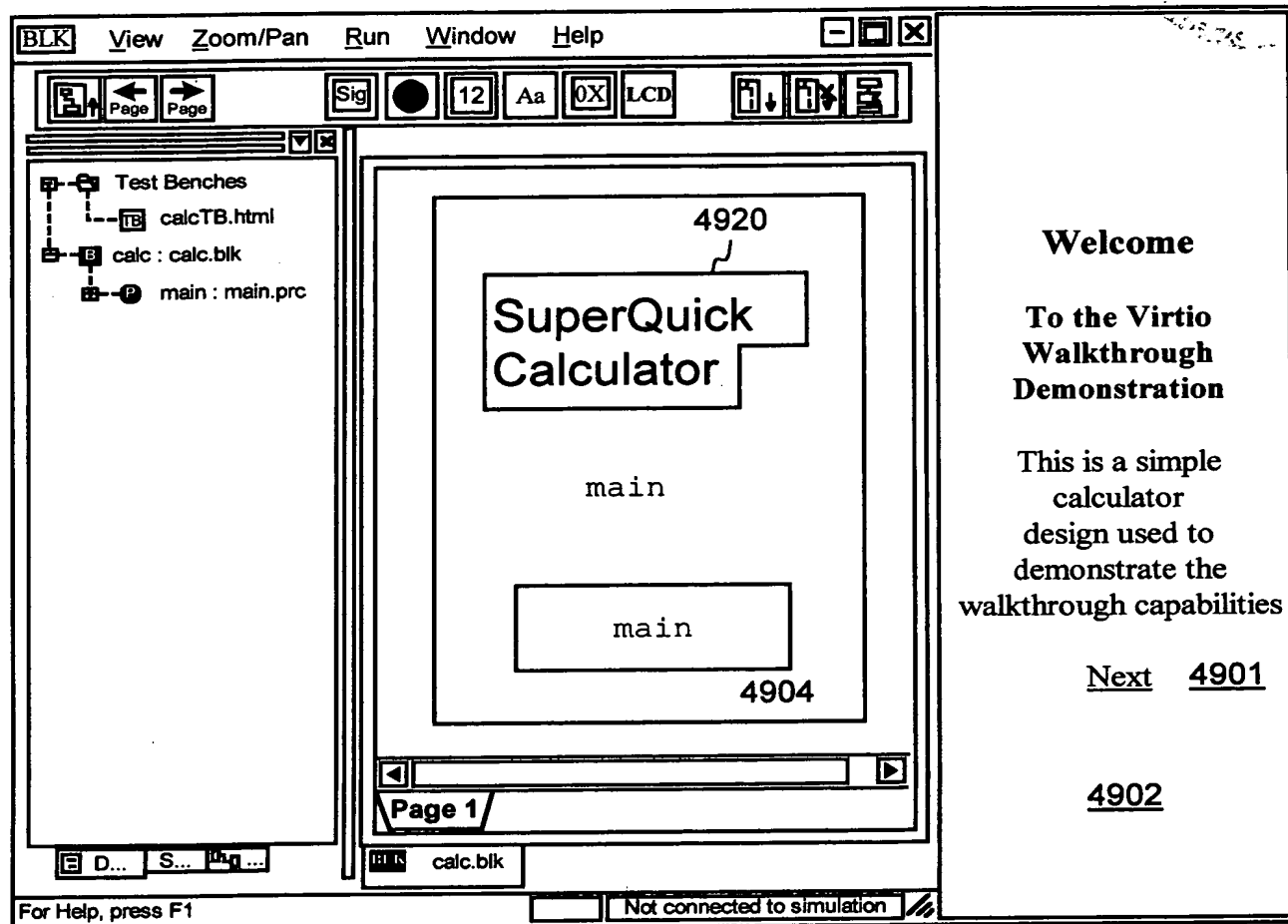


Figure 49A

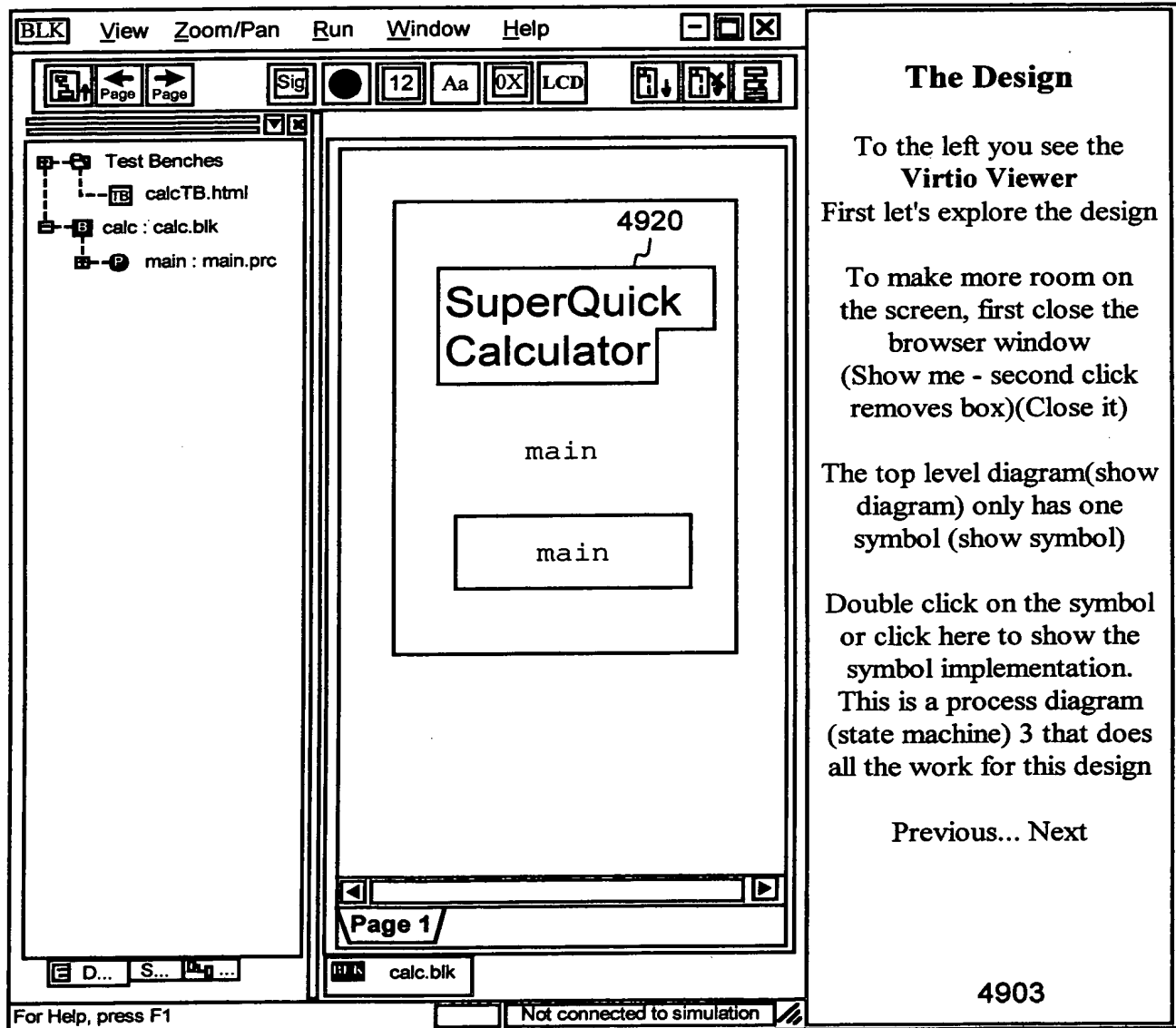


Figure 49B

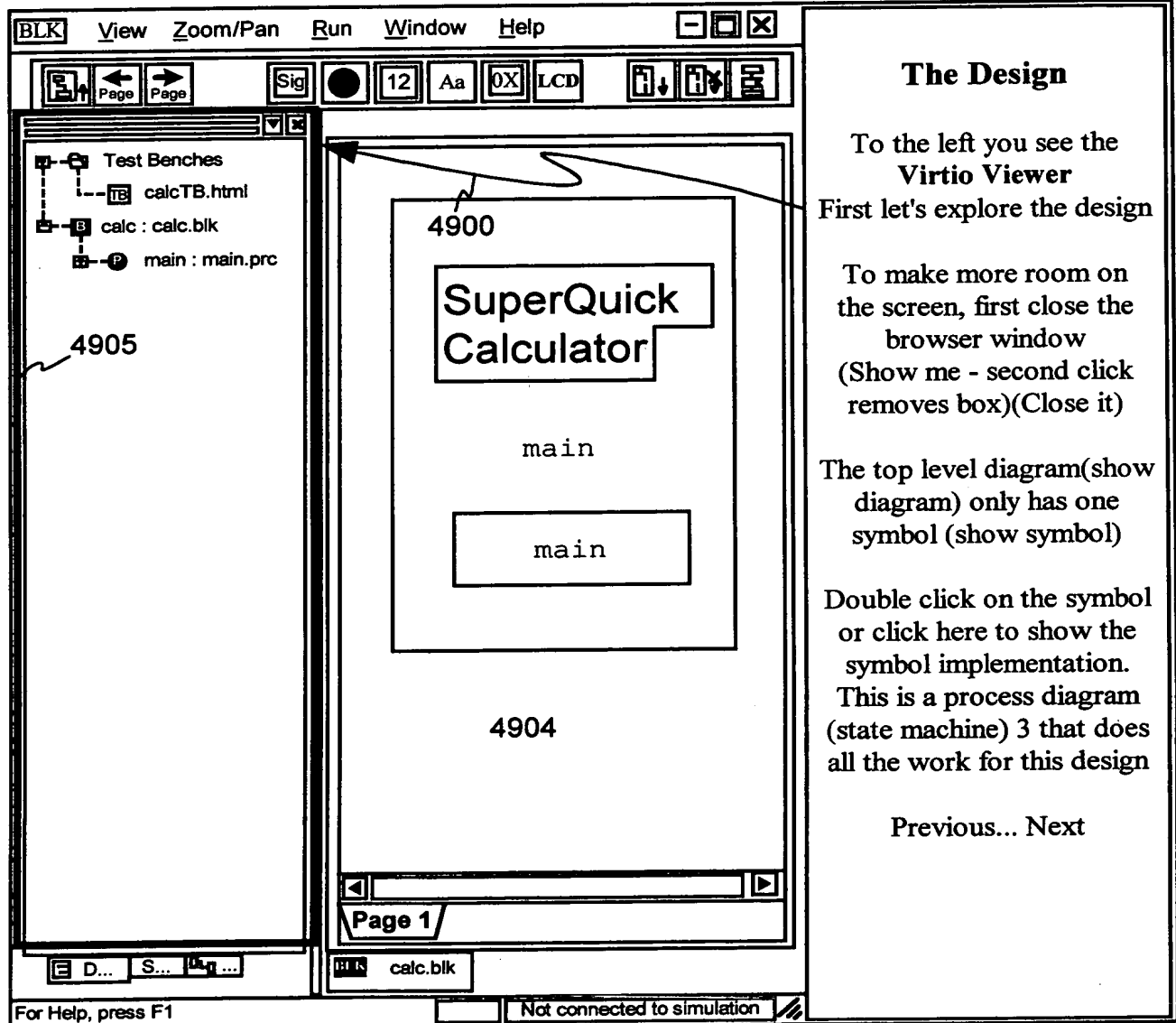
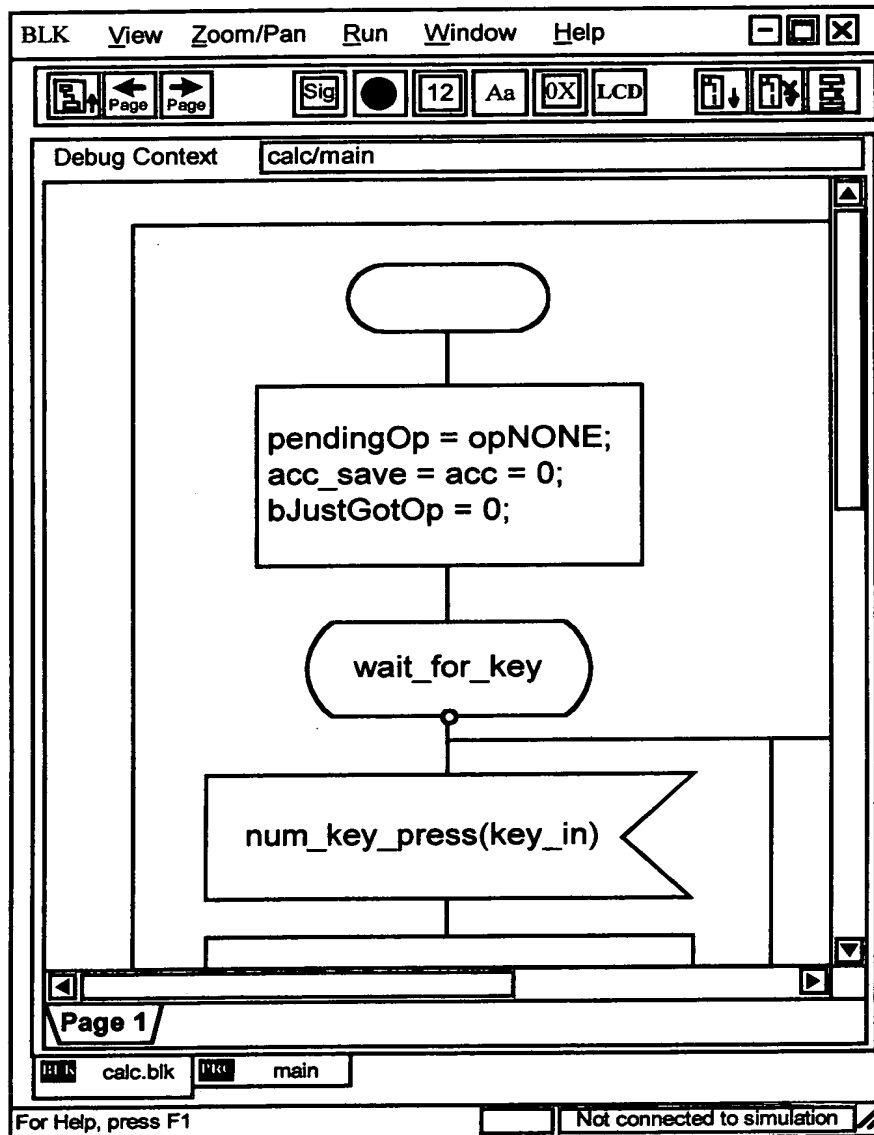


Figure 49C



The Design

To the left you see the
Virtio Viewer
 First let's explore the design

To make more room on
 the screen, first close the
 browser window
 (Show me - second click
 removes box)(Close it)

The top level diagram(show
 diagram) only has one
 symbol (show symbol)

Double click on the symbol
 or click here to show the
 symbol implementation.
 This is a process diagram
 (state machine) 3 that does
 all the work for this design

Previous... Next

Figur 49D

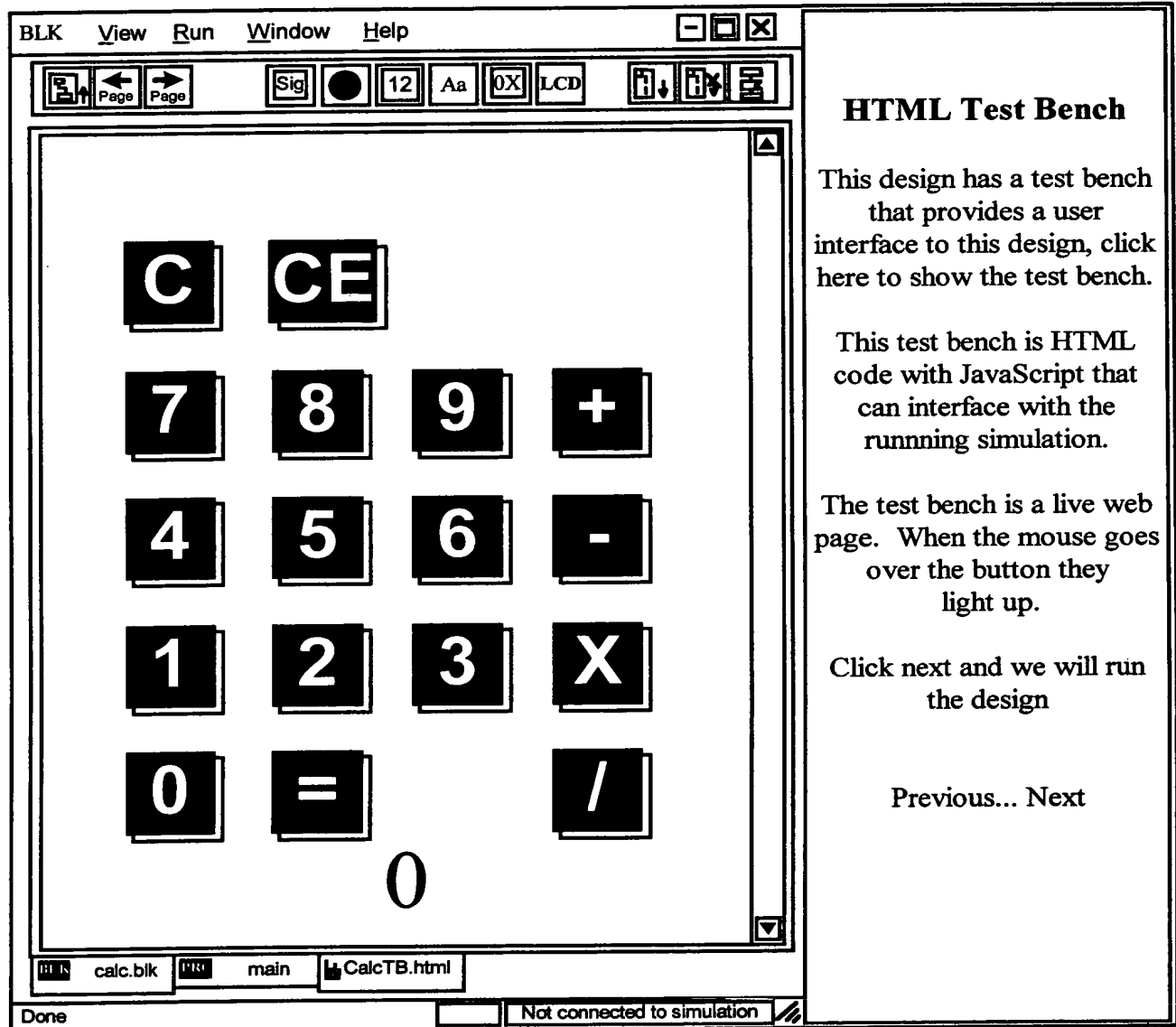
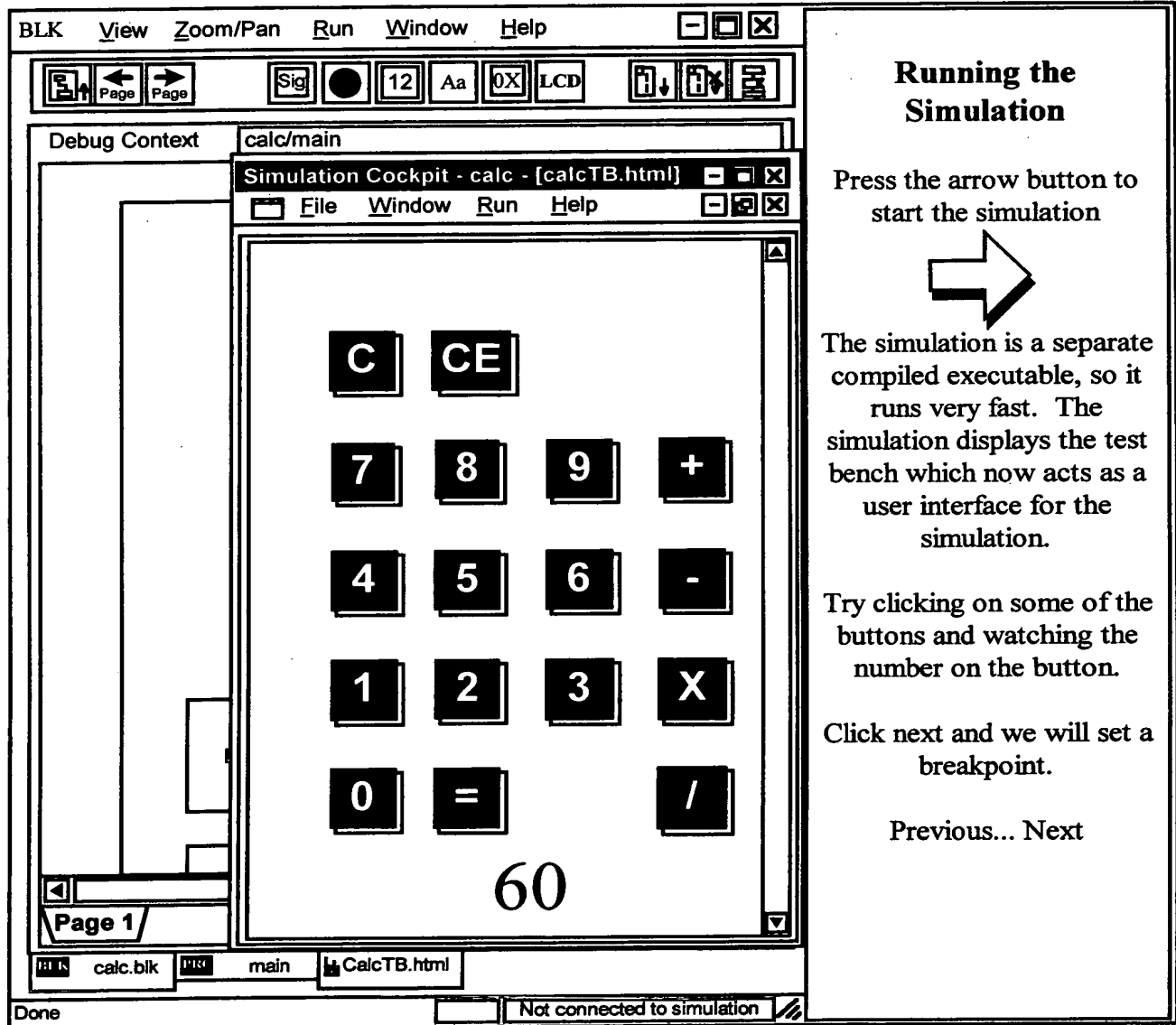
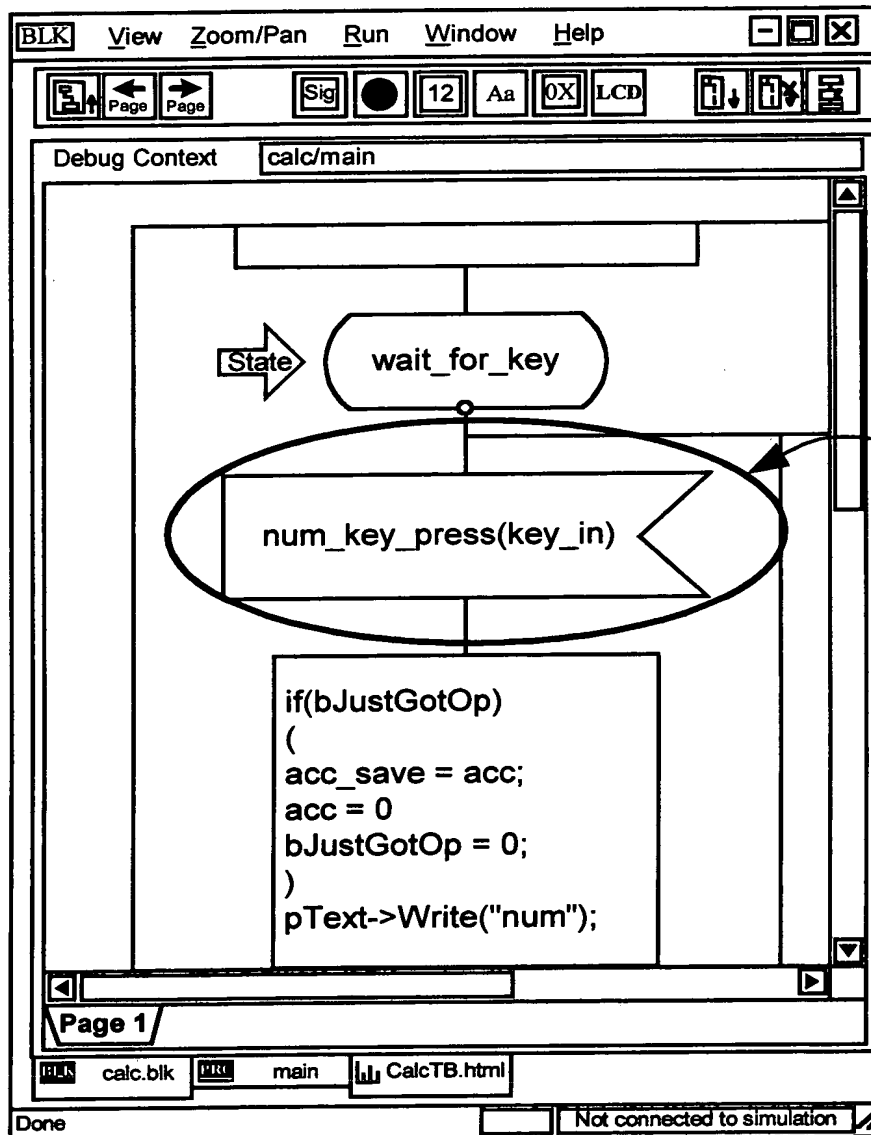


Figure 49E



Figur 49F



Setting a Breakpoint

To set a breakpoint, first check the viewer icon:



to bring the Virtio Viewer to the front

The simulation is still running, it is just behind the viewer now

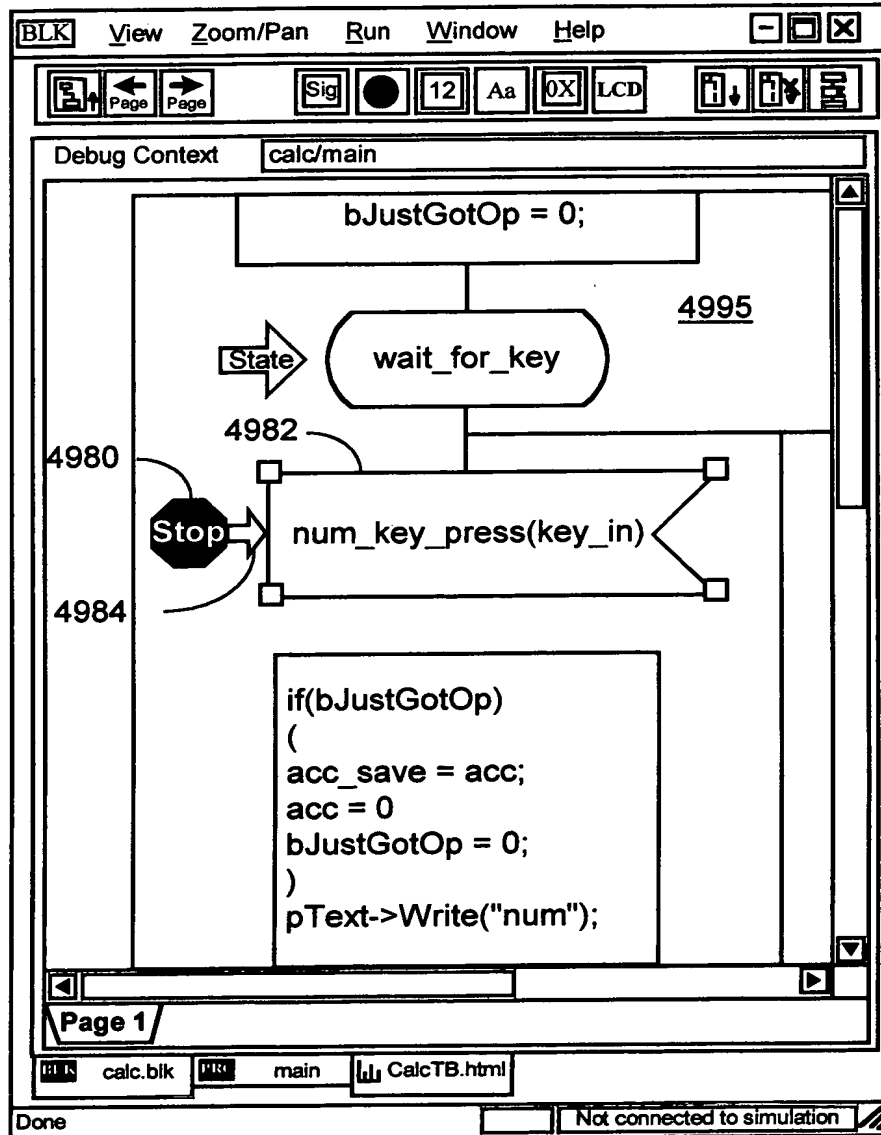
We will set a breakpoint on a Signal In construct. Check here to show it. (the next click will remove the arrow).

Set a breakpoint by right clicking on the Signal In, or by checking this stop icon:




Previous... Next


Figur 49G



Single Stepping

Click this icon  to bring the simulation to the front again.

Now Click on a number key in the simulation so it will hit on a breakpoint

Click here  to bring the Virtio Viewer forward. It should be stopped at the breakpoint.

You can single step from the Viewer menu or toolbar, or this icon:



Single step two times and go to the next page to see how the simulation communicates with the test bench.

Previous... Next

Figur 49H

Simulation to Test Bench Communication

Now the simulation should be stopped on a task block. Click here to show it.

A task block is C++ code that is executed as part of a simulation. This task block modifies the variable acc.

On the test bench (click to show it) the number on the bottom displays the current value of acc.

Now step once more while watching the test bench. That is all it takes to update the test bench!

Previous... Next

Figur 49I

Simulation to Test Bench Communication

Now the simulation should be stopped on a task block. Click here to show it.

A task block is C++ code that is executed as part of a simulation. This task block modifies the variable acc.

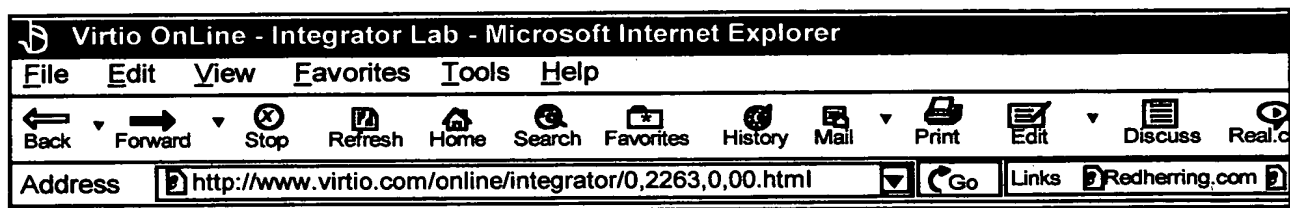
On the test bench (click to show it) the number on the bottom displays the current value of acc.

Now step once more while watching the test bench. That is all it takes to update the test bench!

Previous... Next

Figur 49J

Integrator Lab Screen Snap Shots - On-Line Enablement



[About](#)

[Newsroom](#)

[Technology](#)

[Support](#)

[Virtio Online](#)

[Partners](#)

[Explorer Studio](#) [Integrator Lab](#) [Platform Publisher](#)

[Virtio](#) [Virtio OnLine](#) [Integrator Lab](#)

Integrator Lab

Current Designs		Browse IP		
<u>Design Name</u>	<u>Creation Date</u>	<u>Last Edit Date</u>	<u>Description</u>	<u>Delete Project</u>
<u>test</u>	02-May-01	29-May-01	test	Delete
<u>pcnetlink</u>	04-May-01	25-May-01	asdasd	Delete
<u>clonetutorial</u>	23-May-01	23-May-01	testing cloning of build77	Delete
<u>clonehanoi</u>	25-Apr-01	31-May-01	cloning hanoi	Delete
<u>cloneatlas</u>	25-Apr-01	29-May-01	cloning atlas	Delete

[New Design](#)

[My Invitees](#)

Figure 50

Design Wizard - Setup New Design (Step 1) - Microsoft Internet Explorer

virtio
Integrator Lab

Step 1
Setup New Design
Please fill in the name and description for the new design

Step 2
Browse & Select IP

Step 3
Edit Design

Step 4
Upload Software

Step 5
Run Software

Abort

Setup New Design

Please provide a name and description for your new design

NOTE: Design names must start with a letter and may only contain alpha or numeric characters

Design Name 5110

Description 5115

Start by cloning an existing platform:

Clone 5120

Figure 51

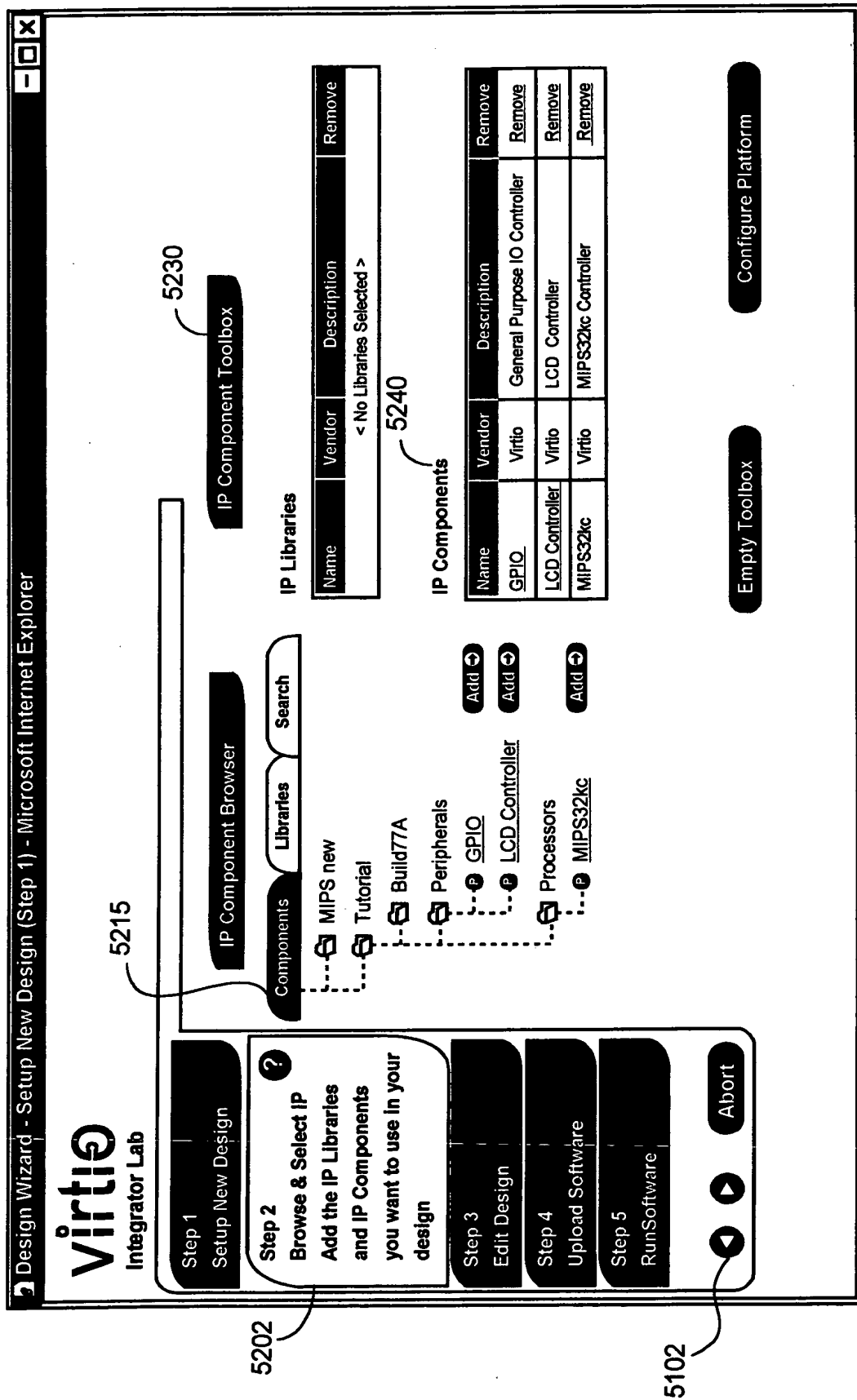


Figure 52



Figure 53

Design Wizard - Upload Embedded Software (Step 4)- Microsoft Internet Explorer

virtio

Integrator Lab

Step 1
Setup New Design

Step 2
Browse & Select IP

Step 3
Edit Design

Step 4
Upload Software
Upload both source
and binary files for test
driving your design

Step 5
RunSoftware

Upload Software

Design : tutorial77a

You can test-drive the hardware design you configured before (step 3) by uploading embedded software to your design repository. In the next step (step 5), you will be able to run & debug on-line the uploaded embedded software.

Please use the Browse button below to select source and binary software files on your local system. Click the Upload button to upload the selected file to your on-line design repository

Note: Either C/C++ source files (extensions *.c, *.C, *.cpp, *.CPP, *.h, *.H, *.inc) or binary executable object files (any extension) for the target platform can be uploaded. To enable source level debugging, you will need provide the source files for all the binaries you decide to upload.

Software development tools and design examples can be found [here](#).

File Name	Date	Size (bytes)	View	Delete
demo.c	May 31 16:06	1005	View	Delete
demoram.x	May 31 16:02	206496		Delete

File to be uploaded

Browse

Upload File

Figure 54

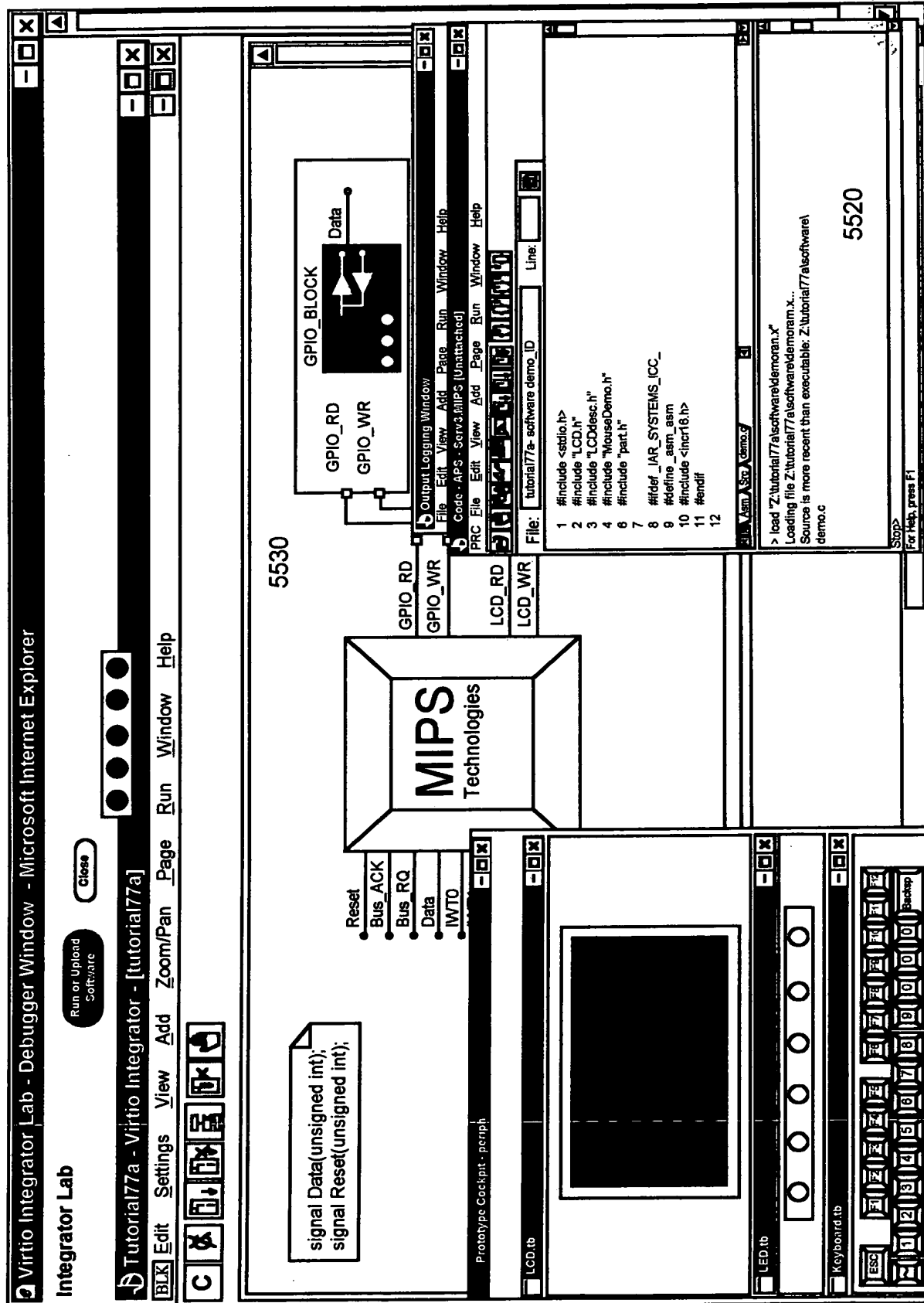


Figure 55